520.43454X00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	HOSOYA, Mutsumi
Serial No.:	Not yet assigned
Filed:	February 2, 2004
Title:	DATA TRANSFER METHOD AND DISK CONTROL UNIT USING IT
Group:	Not yet assigned

LETTER CLAIMING RIGHT OF PRIORITY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

February 2, 2004

Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby claim(s) the right of priority based on Japanese Patent Application No.(s) 2003-353219, filed October 14, 2003.

A certified copy of said Japanese Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP

Carl I. Brundidge
Registration No. 29,621

CIB/alb
Attachment
(703) 312-6600

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年１０月１４日

出願番号
Application Number: 特願２００３－３５３２１９

[ST. 10/C]: [ＪＰ２００３－３５３２１９]

出願人
Applicant(s): 株式会社日立製作所

２００３年１０月２９日

特許庁長官
Commissioner,
Japan Patent Office 今井康夫

出証番号　出証特２００３－３０８９６２７

【書類名】　特許願
【整理番号】　NT03P0585
【提出日】　平成15年10月14日
【あて先】　特許庁長官　殿
【国際特許分類】　G06F 13/00
【発明者】
　【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内
　【氏名】　細谷　睦
【特許出願人】
　【識別番号】　000005108
　【氏名又は名称】　株式会社日立製作所
【代理人】
　【識別番号】　100068504
　【弁理士】
　【氏名又は名称】　小川　勝男
　【電話番号】　03-3661-0071
【選任した代理人】
　【識別番号】　100086656
　【弁理士】
　【氏名又は名称】　田中　恭助
　【電話番号】　03-3661-0071
【選任した代理人】
　【識別番号】　100094352
　【弁理士】
　【氏名又は名称】　佐々木　孝
　【電話番号】　03-3661-0071
【手数料の表示】
　【予納台帳番号】　081423
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　特許請求の範囲　1
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1

【書類名】特許請求の範囲
【請求項１】
イニシエータからターゲットへのデータ転送の際、前記ターゲットで受信されたデータの通信エラーの有無を前記データに付加されているエラーチェックコードを用いて確認し、前記通信エラーの有無を前記ターゲットから前記イニシエータに転送ステータスとして返送し、前記転送ステータスにより前記データ転送の際に転送エラーの発生が判明した場合、前記イニシエータから前記ターゲットに前記データの再送を行うリライアブル転送において、
前記イニシエータと前記ターゲット間の前記データ転送単位である論理レコードのデータ転送方法であって、
前記論理レコードを複数まとめて一括転送し、
前記イニシエータでは、前記一括転送の単位で前記転送ステータスの確認を行い、
前記イニシエータが発行した転送リクエストにより前記論理レコードの転送がなされ、
前記ターゲットでは、前記論理レコードの前記転送リクエストに対応した完了ステータスを予め定められた一括転送条件に合致した前記論理レコードについて、その正常受信が完了した時点で前記ターゲット内に存在する完了キューに通知することを特徴とするデータ転送方法。
【請求項２】
イニシエータからターゲットへのデータ転送の際、前記ターゲットで受信されたデータの通信エラーの有無を前記データに付加されているエラーチェックコードを用いて確認し、前記通信エラーの有無を前記ターゲットから前記イニシエータに転送ステータスとして返送し、前記転送ステータスにより前記データ転送の際に転送エラーの発生が判明した場合、前記イニシエータから前記ターゲットに前記データの再送を行うリライアブル転送において、
前記イニシエータと前記ターゲット間の前記データ転送の単位である論理レコードのデータ転送方法であって、
前記論理レコードを複数まとめて一括転送し、
前記イニシエータが発行した転送リクエストにより前記論理レコードの転送がなされ、それが正常に前記ターゲットに到着した時点で、前記論理レコードの前記転送リクエストに対応した完了ステータスを前記ターゲット内に存在する完了キューに通知するようになし、
前記イニシエータでは、前記一括転送の単位で前記転送ステータスの確認を行い、
前記ターゲットでは、前記一括転送の途中で前記論理レコードの転送エラーを前記エラーチェックコードにより検出した際、前記一括転送の終了まで、当該論理レコードおよびそれ以降の論理レコードの受信を無視し、前記完了ステータスの通知を抑制することを特徴とするデータ転送方法。
【請求項３】
イニシエータからターゲットへのデータ転送の際、前記ターゲットで受信されたデータの通信エラーの有無を前記データに付加されているエラーチェックコードを用いて確認し、前記通信エラーの有無を前記ターゲットから前記イニシエータに転送ステータスとして返送し、前記転送ステータスにより前記データ転送の際に転送エラーの発生が判明した場合、前記イニシエータから前記ターゲットに前記データの再送を行うリライアブル転送において、
前記イニシエータと前記ターゲット間の前記データ転送の単位である論理レコードのデータ転送方法であって、
前記論理レコードを複数まとめて一括転送し、
前記イニシエータでは、前記一括転送の単位で前記転送ステータスの確認を行い、
前記イニシエータが発行した転送リクエストにより前記論理レコードの転送が正常に前記ターゲットに到着した時点で、前記論理レコードの前記転送リクエストに対応した完了ステータスを前記ターゲット内に存在する完了キューに通知し、

前記ターゲットでは、前記一括転送の途中で前記論理レコードの転送エラーを前記エラーチェックコードにより検出した際、前記一括転送の終了まで、当該論理レコード及び、それ以降の論理レコードのうち一括転送条件フィールドで指定されている論理レコードの受信を無視し、前記完了ステータスの通知を抑制することを特徴とするデータ転送方法。

【請求項４】

前記イニシエータと前記ターゲットとの間の前記一括転送において、

前記ターゲットでは転送エラーが検出された最初の論理レコードのＩＤを、前記一括転送の単位ごとに確認の行われる前記転送ステータスの中に含め、

前記イニシエータでは前記転送ステータスをもとに、転送エラーの発生した論理レコードから再送を開始することを特徴とする請求項１乃至請求項３のいずれかに記載のデータ転送方法。

【請求項５】

前記イニシエータと前記ターゲットとの間の前記一括転送において、

前記ターゲットでは転送エラーが検出された論理レコードのＩＤのリストを、前記一括転送の単位ごとに確認が行われる前記転送ステータスの中に含め、

前記イニシエータでは前記リストをもとに、転送エラーが発生した論理レコードの再送を行うことを特徴とする請求項１乃至請求項３のいずれかに記載のデータ転送方法。

【請求項６】

前記一括転送の途中で、前記イニシエータ、もしくは、前記ターゲットからキャンセルリクエストを発行して、前記一括転送を中止することが可能であることを特徴とする請求項１乃至請求項５のいずれかに記載のデータ転送方法。

【請求項７】

ホストコンピュータとのインターフェースを有する複数のホストインターフェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部とを有し、前記ホストインターフェース部は、前記ホストコンピュータに対しリード／ライトされるデータを一時的に格納するメモリを有し、前記メモリと前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は前記磁気ディスク装置に対しリード／ライトされるデータを一時的に格納するキャッシュメモリを有し、前記メモリと前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、

前記複数のホストインターフェース部と前記複数のディスクインターフェース部との間の転送に請求項１乃至請求項６のいずれかに記載のデータ転送方法が適用されていることを特徴とするディスク制御装置。

【請求項８】

ホストコンピュータとのインターフェースを有する複数のホストインターフェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部とを有し、前記ホストインターフェース部は、前記ホストコンピュータに対しリード／ライトされるデータを一時的に格納するメモリを有し、前記メモリと前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は前記磁気ディスク装置に対しリード／ライトされるデータを一時的に格納するキャッシュメモリを有し、前記メモリと前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、

前記複数のホストインターフェース部間の転送に請求項１乃至請求項６のいずれかに記載のデータ転送方法が適用されていることを特徴とするディスク制御装置。

【請求項９】

ホストコンピュータとのインターフェースを有する複数のホストインターフェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部とを有し、前記ホストインターフェース部は、前記ホストコンピュータに対しリード／ライトされるデータを一時的に格納するメモリを有し、前記メモリと前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は前記磁気ディスク装置に対しリード／ライトされるデータを一時的に格納するキャッシュメモリを有し、前記メモリと前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、

前記複数のディスクインターフェース部間の転送に請求項１乃至請求項６のいずれかに記載のデータ転送方法が適用されていることを特徴とするディスク制御装置。

【請求項１０】

ホストコンピュータとのインターフェースを有する複数のホストインターフェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部と、複数のメモリ部と、複数のプロッセサ部とを有し、前記ホストインターフェース部は、前記プロッセサ部と連携して前記メモリ部と前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は、前記プロセッサ部と連携して前記メモリ部と前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、

前記複数のホストインターフェース部あるいは前記複数のディスクインターフェース部と前記メモリ部との間の転送に請求項１乃至請求項６のいずれかに記載のデータ転送方法が適用されていることを特徴とするディスク制御装置。

【請求項１１】

ホストコンピュータとのインターフェースを有する複数のホストインターフェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部と，複数のメモリ部と、複数のプロセッサー部とを有し、前記ホストインターフェース部は、前記プロッセサ部と連携して前記メモリ部と前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は、前記プロセッサ部と連携して前記メモリ部と前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、

前記複数のホストインターフェース部あるいは前記複数のディスクインターフェース部と前記プロセッサー部間の転送に請求項１乃至請求項６のいずれかに記載のデータ転送方法が適用されていることを特徴とするディスク制御装置。

【請求項１２】

ホストコンピュータとのインターフェースを有する複数のホストインターフェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部と，複数のメモリ部と、複数のプロセッサー部とを有し、前記ホストインターフェース部は、前記プロッセサ部と連携して前記メモリ部と前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は、前記プロセッサ部と連携して前記メモリ部と前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、

前記複数のメモリ部とプロセッサー部間の転送に請求項１乃至請求項６のいずれかに記載のデータ転送方法が適用されていることを特徴とするディスク制御装置。

【請求項１３】

ホストコンピュータとのインターフェースを有する複数のホストインターフェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部と，複数のメモリ部と、複数のプロセッサー部とを有し、前記ホストインターフェース部は、前記プロッセサ部と連携して前記メモリ部と前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は、前記プロセッサ部と連携して前記メモリ部と前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、

前記複数のメモリ部間の転送に請求項１乃至請求項６のいずれかに記載のデータ転送方法が適用されていることを特徴とするディスク制御装置。

【請求項１４】

ホストコンピュータとのインターフェースを有する複数のホストインターフェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部と，複数のメモリ部と、複数のプロセッサー部とを有し、前記ホストインターフェース部は、前記プロッセサ部と連携して前記メモリ部と前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は、前記プロセッサ部と連携して前記メモリ部と前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、

前記複数のプロセッサ部間の転送に請求項１乃至請求項６のいずれかに記載のデータ転送方法が適用されていることを特徴とするディスク制御装置。

【書類名】明細書
【発明の名称】データ転送方法、および、これを用いたディスク制御装置
【技術分野】
【０００１】
本発明は、ネットワークを介して高信頼かつ高速にデータを転送するためのデータ転送方法と複数の磁気ディスク装置を制御するためのディスク制御装置に関する。
【背景技術】
【０００２】
情報通信インフラの高度化に伴い、情報通信システムに求められる処理性能は益々高まってきている。とくに最近では、集積ＬＳＩの微細化が進展した結果、ＬＳＩ間のデータ転送性能が、システム性能を左右する状況となっている。このため、ＩＯシステムの高性能化・多機能化に対する研究が精力的に進められており、転送速度を向上するとともに、多様なトランスポート機能を持った通信プロトコル・エンジンが開発されるようになってきた。
【０００３】
例えば、InfiniBand Architecture Specification Release 1.0aで規定されているInfiniBand転送方式は、送信キュー（ＳＱ）および受信キュー（ＲＱ）で構成されるキューペア（ＱＰ）と、キューペアに積まれたリクエスト処理完了時点で完了ステータスの通知される完了キュー（ＣＱ）とで、アプリケーションプロセスとＩＯシステムとのインターフェースがとられるデータ転送方法となっている。その様子を図4で説明する。
【０００４】
プロセス５１とプロセス５２は、それぞれ２つのキューペアを用いて通信している。プロセス５１は、送信キュー１１と受信キュー２１からなるキューペア４１と送信キュー１２と受信キュー２２からなるキューペア４２を持っている。同様に、プロセス５２は、送信キュー１３と受信キュー２３からなるキューペア４３と送信キュー１４と受信キュー２４からなるキューペア４４を持っている。完了キュー３１には、キューペア４１およびキューペア４２の完了ステータスが格納され、完了キュー３２には、キューペア４３およびキューペア４４の完了ステータスが格納される。
【０００５】
送信キューの各エントリには、転送リクエストがおかれる。この転送リクエストにより転送されるデータの単位を論理レコードと呼ぶ。受信キューの各エントリには、受信バッファへのポインタが格納される。送信キュー１２の転送リクエストは、プロセスバッファ７１内のレコードバッファ８１へのポインタを有し、送信キュー１４の転送リクエストは、プロセスバッファ７２内のレコードバッファ８２へのポインタを有している。同様に、受信キュー２２、２４は、レコードバッファ８１、８２へのポインタが格納されている。
【０００６】
通信を行う２つのキューペア間では、送信キューと受信キュー間で接続が張られる。送信キュー１２は受信キュー２４と、送信キュー１４は受信キュー２２と接続がなされている。このとき、送信キュー１２に積まれた転送リクエストが処理されると、レコードバッファ８１に格納されていた論理レコードが受信キュー２４で指定されているレコードバッファ８２へ転送される。正常な転送が完了した時点で、受信キュー２４からは完了キュー３２に完了ステータスが通知され、送信キュー１２からは完了キュー３１へ完了ステータスが通知される。
【０００７】
これらのキューペアや完了キューの処理は、ホストチャネルアダプタ（ＨＣＡ）とよばれるハードウェアで実行される。その構成例を図５に示した。ＨＣＡは、受信ポート６１３と送信ポート６２３、受信側のリンク層論理６３１、トランスポート層論理６３２、プロセッサ６３３、送信側のリンク層論理６４１、トランスポート層論理６４２、プロセッサ６４３、メモリ６５０、および、接続インターフェース６６０を有している。ＨＣＡとアプリケーションプロセスとの通信は、接続インターフェース、および、メモリを介して

行われる。受信側と送信側は、並行して動作することができ、各プロセッサと、リンク層・トランスポート層論理で、高機能プロトコル処理を高速に実行することができるようになっている。

【０００８】

２つのＨＣＡ間で単純転送リクエストが処理される様子を図６で説明する。ＨＣＡ１側のプロセスバッファ７３内のレコードバッファ８３をＨＣＡ２側のプロセスバッファ７４内のレコードバッファ８４に転送する場合を考える。ＨＣＡ１では、レコードバッファ８３を送信に適したサイズに分解し、それぞれ適当なヘッダとエラーチェックコード（ＣＲＣ）を付加してパケット４０１－４０３として転送する。ＨＣＡ２では、受信パケットにエラーがないかＣＲＣで確認し、エラーを検出した場合、ＨＣＡ１にその旨ＮＡＫ（Negative AcKnowledgement）で通知する。ＨＣＡ１では、ＮＡＫが返されたパケットについては再送を行う。ＨＣＡ２では、すべてのパケットが正常に受け取れた時点で、論理レコードの再構築を行い、レコードバッファ８４に格納して、完了キュー３４に完了ステータスを通知するとともに、ＨＣＡ１に受信完了を知らせる。ＨＣＡ１では、ＨＣＡ２の受信完了通知を受けて、完了キュー３３に転送完了ステータスを通知することで、一連の転送リクエスト処理が完了する。

【０００９】

別の例として、２つのＨＣＡ間でのＲＤＭＡ転送リクエストが処理される様子を図７で説明する。ＲＤＭＡ転送では、イニシエータのアプリケーションメモリ空間７５内のＲＤＭＡ転送空間８５をターゲットのアプリケーションメモリ空間７６内のＲＤＭＡ転送空間８６に転送する。ＲＤＭＡ転送では、転送先のアプリケーションメモリ空間内に直接転送を行うので、転送先のメモリアドレス情報を付加する必要がある。それ以外は、単純転送リクエストと同様の動作をする。ＨＣＡ１ではＲＤＭＡ転送空間８５を適当なサイズでパケットに構成し、順次ＨＣＡ２に転送する。ＨＣＡ２では、受信パケットをＲＤＭＡ転送空間８６内の決められた場所に格納し、必要ならパケットの再送処理をして、空間全体の再構築を行う。ＨＣＡ２では、すべてのパケットを正常に受け取った時点で、完了キュー３６に完了ステータスを通知するとともに、ＨＣＡ１に受信完了を知らせる。ＨＣＡ１では、ＨＣＡ２の受信完了通知を受けて、完了キュー３５に転送完了ステータスを通知することで、一連の転送リクエスト処理が完了する。

【００１０】

ここまで説明したデータ転送方法は、転送エラーの無いことを保証するリライアブルデータ転送方法であり、広く一般的に使われている基本的なものである。従来のリライアブルデータ転送方法には、以下の２つの特徴がある。

１．　ターゲットでは、転送リクエスト単位である論理レコード全体にエラーの無いことを確認してから完了ステータスを通知する。

２．　イニシエータでは、ターゲットからの論理レコード全体の正常転送完了ステータス通知を確認してから次の論理レコードの転送を開始する。

【００１１】

上記特徴について、図２、および、図３を用いて説明する。図２では、ＨＣＡ１側のアプリケーション（ＡＰ）１から転送リクエスト１２１によって論理レコード２２１の転送を開始している。ＨＣＡ２で転送エラーを検出した場合には再送を行う。ＨＣＡ２は、正常に論理レコードの受信を完了した時点で、アプリケーション２の完了キューに完了ステータス３２１を通知する。完了ステータス３２１を受けて、アプリケーション２は、論理レコード２２１を使った処理７２１を開始することができる。論理レコード２２１の正常受信を完了したＨＣＡ２は、受信完了をＨＣＡ１に知らせ、ＨＣＡ１はアプリケーション１の完了キューに転送完了ステータス３６１を通知する。この例で明らかなように、ターゲット側のアプリケーション２は、論理レコード２２１全体の受信が完了してから完了ステータス３２１を受け取る。また、イニシエータ側のアプリケーション１は、論理レコード２２１全体の受信がＨＣＡ２で完了した後、次の論理レコードの転送リクエストを開始することができる。

【００１２】
図３では、ＨＣＡ１側のアプリケーション１から転送リクエスト１３１によってＲＤＭＡ転送を開始している。この場合、ＲＤＭＡ転送空間全体がひとつの論理レコードとなる。ＨＣＡ２では、複数に分割されたパケットを受信しながら、必要に応じて再送要求を行う。すべてのパケット転送（論理レコード全体の転送）が正常に終了した時点で、ＨＣＡ２側のアプリケーション２の完了キューに完了ステータス３３１を通知する。完了ステータス３３１を受けて、アプリケーション２は、転送された論理レコードであるＲＤＭＡ転送空間を使った処理７３１を開始することができる。論理レコードの正常受信を完了したＨＣＡ２は、受信完了をＨＣＡ１に知らせ、ＨＣＡ１はアプリケーション１（の完了キュー）に転送完了ステータス３７１を通知する。
【００１３】
この例でも明らかなように、ターゲット側のアプリケーション２は、論理レコード（ＲＤＭＡ転送空間）全体の受信が完了してから完了ステータス３３１を受け取る。また、イニシエータ側のアプリケーション１は、論理レコード（ＲＤＭＡ転送空間）全体の受信がＨＣＡ２で完了した後、次の論理レコードの転送リクエストを開始することができる。
【００１４】
このように従来のリライアブルデータ転送方法では、転送エラーを回避するための仕組みとして、上記２つの基本的な特徴をもつ必要があった。なお、特開平８－１７９９９９に開示されている従来のリライアブルデータ転送方法のように、エラー発生以前の転送データを保証する方式も知られているが、上記２つの特徴を満たす必要のあることに変わりはない。
【００１５】
【特許文献１】特開平８－１７９９９９号公報
【００１６】
【非特許文献１】InfiniBand Architecture Specification Release 1.0a
【発明の開示】
【発明が解決しようとする課題】
【００１７】
従来のリライアブルデータ転送方法は、転送エラーを回避するために上記２つの特徴を利用しているのであるが、そのために、以下の課題をもつ。
まず、「イニシエータがターゲットでの論理レコード全体の正常受信完了を確認する必要がある」という特徴のため、イニシエータからターゲットへの論理レコード転送に要する時間以外に、ターゲットからイニシエータに転送完了を通知するための時間がオーバーヘッドとしてかかってしまう。このオーバヘッド時間は、論理レコードのレコード長が短い時に顕著となり、通信路の効率を著しく低下させる。とくに、ＩＯシステムの高機能化に伴ってターゲットでの処理タスクは増加しており、論理レコードの転送完了ステータス通知に要する時間は増大傾向にある。論理レコードの転送時間は、転送速度の向上にしたがって短縮されているので、イニシエータからの転送完了通知オーバーヘッドが相対的に増大して転送効率が悪化しており、その改善が課題となっている。
【００１８】
次に、「ターゲットでは、受信した論理レコード全体にエラーの無いことを確認してから完了ステータスを通知する」という特徴のため、たとえ論理レコードの途中まで正常に受信ができたとしても、それをターゲットのアプリケーションで認識して有効に活用することができず、論理レコード全体の受信が完了するまで、受信論理レコードを利用したアプリケーション処理の開始を遅延させる必要がある。この処理開始までの遅延時間は、論理レコードのレコード長が長いときに顕著となり、アプリケーションの処理効率を低下させる。レコード長が長ければ、それだけ転送エラーの発生率が上昇し、転送エラーによる再送処理が行われると、上記遅延時間はさらに増加し、処理効率は低下する。大きなレコード長の転送や、転送エラーの発生時にも、アプリケーションの処理効率を低下させないことが、別の課題となっている。

【００１９】
本発明の目的は、上記従来技術の欠点を改善し、高い転送効率と高いアプリケーション処理効率を同時に実現するデータ転送方法を提供することにある。より具体的には、ターゲットでの受信完了通知に要する時間とイニシエータへの転送完了通知に要する時間を実効的に削減するデータ転送方法およびかかるデータ転送方法を用いたディスク制御装置を提供することにある。
【課題を解決するための手段】
【００２０】
上記目的を達成するため、本発明では、イニシエータからターゲットへのデータ転送の際、ターゲットで受信されたデータの通信エラーの有無をデータに付加されているエラーチェックコードを用いて確認し、通信エラーの有無をターゲットからイニシエータに転送ステータスとして返送し、転送ステータスによりデータ転送の際に転送エラーの発生が判明した場合、イニシエータからターゲットにデータの再送を行うリライアブル転送において、イニシエータとターゲット間のデータ転送単位である論理レコードの転送プロトコルにおいて、論理レコードを複数まとめて一括転送し、イニシエータでは、一括転送の単位で転送ステータスの確認を行い、イニシエータが発行した転送リクエストにより論理レコードの転送がなされ、ターゲットでは、転送リクエストに対応した完了ステータスを予め定められた一括転送条件に合致した論理レコードについて、その正常受信が完了した時点でターゲット内に存在する完了キューに通知するようにした。
【００２１】
また、ターゲットでは、一括転送の途中で論理レコードの転送エラーをエラーチェックコードにより検出した際、一括転送の終了まで、当該論理レコードおよびそれ以降の論理レコードの受信を無視し、前記完了ステータスの通知を抑制するようにした。
【００２２】
さらに、ターゲットでは、一括転送の途中で論理レコードの転送エラーをエラーチェックコードにより検出した際、一括転送の終了まで、当該論理レコード及び、それ以降の論理レコードのうち一括転送条件フィールドで指定されている論理レコードの受信を無視し、完了ステータスの通知を抑制するようにした。
【００２３】
さらにまた、ターゲットでは転送エラーが検出された最初の論理レコードのＩＤを、一括転送の単位ごとに確認の行われる前記転送ステータスの中に含め、イニシエータでは転送ステータスをもとに、転送エラーの発生した論理レコードから再送を開始するようにした。
【００２４】
また、一括転送の途中で、イニシエータ、もしくは、ターゲットからキャンセルリクエストを発行して、一括転送を中止することが可能であるようにした。
【００２５】
さらに、ホストコンピュータとのインターフェースを有する複数のホストインターフェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部とを有し、前記ホストインターフェース部は、前記ホストコンピュータに対しリード／ライトされるデータを一時的に格納するメモリを有し、前記メモリと前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は前記磁気ディスク装置に対しリード／ライトされるデータを一時的に格納するキャッシュメモリを有し、前記メモリと前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、複数のホストインターフェース部と複数のディスクインターフェース部との間の転送、複数のホストインターフェース部間の転送、あるいは　複数のホストインターフェース部もしくは複数のディスクインターフェース部と前記メモリ部との間の転送に上述のデータ転送方法を適用するようにした。
【００２６】
さらにまた、ホストコンピュータとのインターフェースを有する複数のホストインター

フェース部と、磁気ディスク装置とのインターフェースを有する複数のディスクインターフェース部と，複数のメモリ部と、複数のプロセッサー部とを有し、前記ホストインターフェース部は、前記プロッセサ部と連携して前記メモリ部と前記ホストコンピュータとの間でデータ転送を実行し、前記ディスクインターフェース部は、前記プロセッサ部と連携して前記メモリ部と前記磁気ディスク装置との間でデータ転送を実行するディスク制御装置において、　前記複数のホストインターフェース部あるいは前記複数のディスクインターフェース部と前記プロセッサー部間の転送、複数のメモリ部とプロセッサー部間の転送、　複数のメモリ部間の転送、あるいは、複数のプロセッサ部間の転送に上述のデータ転送方法を適用するようにした。

【発明の効果】

【００２７】

以上説明したように、本発明のデータ転送方法では、一括転送中には、各論理レコードの転送リクエストが連続して処理され、イニシエータは、ターゲットからの個別の論理レコード転送完了ステータスを待たない。そのため、個別の論理レコード転送完了通知に要する時間が不要となり、転送効率を大幅に向上させることが可能となる。

【００２８】

本発明によれば、一括転送条件に合致した正常受信の論理レコードについては、当該論理レコードの受信完了時点で完了ステータスがアプリケーションに通知される。そのため、多数の論理レコードを一括で転送した場合でも、一括転送全体が完了するのを待たずに、個別の論理レコードの受信が完了した時点で、対応するアプリケーションの処理を開始することができ、アプリケーションの処理効率を改善することが可能となる。とくに、一括転送中に受信エラーが発生した場合についても、エラー発生以前に正常受信した論理レコードについては、その受信完了時点で、アプリケーションの実行を開始できる。

【００２９】

本発明によれば、一括転送における一括転送条件をさらに詳細に規定することができ、一括転送中に転送エラーが生じた場合でも、データ依存関係がなく、到着順序で問題を生じる可能性の無い論理レコードについては、受信動作を継続することが可能となる。そのため、アプリケーションの処理効率を高めることができ、再送レコード数を減らすことでより効率のよい転送を実現することができる。

【００３０】

本発明に拠れば、再送の必要な論理レコードについてのみ再送処理を行うことで、再送処理に要する時間を最小限に抑えることができる。したがって、再送レコード数を減らすことでより効率のよい転送を実現することができる。

【００３１】

本発明に拠れば、一括転送の途中で転送エラー発生などにより、その後の一括転送の継続が不要になった場合、当該一括転送を途中で中止することが可能となる。そのため、不要な論理レコードの転送を抑制することで、より効率のよい転送を実現することができる。

【００３２】

また、本発明のディスク制御装置では、ホストコンピュータ部とディスクインターフェース部の転送効率を大幅に改善することができる。特に、ディスク制御装置内ネットワークの転送効率の向上と、ホストインターフェース部、および、ディスクインターフェース部からのＩＯリクエストに対するレスポンス時間の短縮が可能となり、ディスク制御装置の処理性能を高めることができる効果を奏する。

【発明を実施するための最良の形態】

【００３３】

以下、本発明の実施例を図面を用いて詳述する。

《実施例１》

図１及び図８に、本発明の一実施例を示す。本発明のデータ転送方法においては、複数の論理レコードをまとめて一括転送を行う。そのため、図８に示すように、イニシエータ

の送信キューに複数の転送リクエストが連続して格納される。各転送リクエストには、論理レコードの転送ごとに一意に定まる転送ＩＤと、転送リクエストの動作を規定する転送命令コードと、転送レコードバッファの先頭アドレスおよびそのバッファ長を含んでいる。図８では、転送リクエスト１０１がレコードバッファ８９の単純転送、転送リクエスト１０５がＲＤＭＡ転送領域８７のＲＤＭＡ転送を示しており、これらが一括転送リクエストとして定義されている。

【００３４】

また、図８には、論理レコードの受信完了を通知するために用いられるターゲットの完了キュー３９とそこに格納される(論理レコードの)完了ステータス３００の構成も示してある。完了ステータス３００は、論理レコードの転送ＩＤと対応するステータスＩＤと、その完了ステータスコードが含まれている。

【００３５】

この一括転送リクエストが処理される様子の一例を図１に示した。ここでは、アプリケーション１がＨＣＡ１とＨＣＡ２を介してアプリケーション２に転送リクエスト１０１－１０５を発行している。ＨＣＡ１では、転送リクエスト１０１によって、論理レコード２０１の転送を開始する。ＨＣＡ２では、論理レコード２０１の正常受信完了を確認した後、アプリケーション２の完了キューに対して完了ステータス３０１を通知している。アプリケーション２では、完了ステータス３０１を受けて、論理レコード２０１に対する処理動作７０１を開始することができる。

【００３６】

本発明の高速データ転送方法では、従来方式と異なって、論理レコード１０２の転送を開始するのに、ＨＣＡ２からの論理レコード１０１の転送完了ステータス通知を待たない。ＨＣＡ１では論理レコード１０１の転送が終了した後、ただちに論理レコード１０２の転送を開始する。また、ＨＣＡ２では、受信した論理レコード１０２にエラーが含まれていた場合、従来方式のように論理レコード１０２の再送要求処理を行うのではなく、一括転送完了ステータスに論理レコード１０２の受信エラーを記録した後、次の論理レコード１０３の処理に移る。一旦、転送リクエスト１０５までのすべての一括転送リクエストが処理された時点で、ＨＣＡ２はまとめて、一括転送完了ステータス３５１をＨＣＡ１に返送する。これを受けて、ＨＣＡ1では、ＨＣＡ２で正常受信の完了していない論理レコードの再送を行う。ＨＣＡ２では、再送によって正常に受信できたレコード２１２－２１５については、完了キューへの完了ステータス３１２－３１５の通知を行い、アプリケーション２では対応する処理７１２－７１５を開始する。一括転送完了ステータス３５２によって、すべての一括転送リクエストの正常転送完了ステータスが通知されることで、一連の一括転送が終了する。

【００３７】

本実施例によれば、一括転送中には、各論理レコードの転送リクエストが連続して処理され、イニシエータは、ターゲットからの個別の論理レコード転送完了ステータス通知を待たない。そのため、個別の論理レコード転送完了通知に要する時間が不要となり、転送効率を大幅に向上させることが可能となる。

また、本実施例によれば、一括転送条件に合致した正常受信の論理レコードについては、当該論理レコードの受信完了時点で完了ステータスがアプリケーションに通知される。そのため、多数の論理レコードを一括で転送した場合でも、一括転送全体の完了を待たずに、個別の論理レコード受信の完了した時点で、対応するアプリケーション処理を開始することができ、アプリケーション処理効率を改善することが可能となる。とくに、一括転送中に転送エラーの発生した場合についても、エラー発生以前に正常受信した論理レコードについては、その受信完了時点で、アプリケーションの実行を開始することができる。

【００３８】

《実施例２》

本発明のデータ転送方法では、一括転送の論理レコードを受信する際、一括転送条件に合致したものだけを選択的に受信する。これにより、一括転送内の論理レコード間で依存

関係があっても、受信レコードの時系列順序を保証することが可能となる。例えば、ある論理レコードに受信エラーが含まれていた場合、一括転送中のそれ以降の論理レコードを無視してしまう方法を実施することができる。この方法により、すべての一括転送リクエストを連続して処理した後に、まとめて再送処理をおこなう本発明の方式では、受信論理レコードの依存関係に問題を生じることがなく、従来方式のようにイニシエータでターゲットからの転送完了通知を待つ必要がないという利点がある。

【００３９】

《実施例３》

本発明のデータ転送方法で用いられる一括転送条件について、図９を用いてさらに詳しい説明をおこなう。図９は、本発明のデータ転送方法で使用されるパケットの構造を示したものである。ルーティングに関する情報をもったルーディングヘッダ４４１と、トランスポート処理に関する情報の入ったトランスポートヘッダ４４２と、論理レコード情報の入るペイロード４４３とエラーチェックコードであるＣＲＣ４４４とからなる。ルーディングヘッダは、イニシエータ・ターゲットの宛先アドレスと、パケットの優先度情報、パケット長を含んでいる。トランスポートヘッダには、転送処理を規定する処理動作コードと宛先のキューペア番号、パケット順序番号と、一括転送の動作を規定する一括転送フラグ４５０、一括転送条件フィールド４５１が含まれている。

【００４０】

一括転送フラグ４５０は、当該パケットが一括転送されていることを示すものであり、このフラグを確認することで、一括転送中なのかどうかを判断することができる。一括転送条件フィールド４５１は、一括転送中の各論理レコードのデータ依存関係を示したものである。各論理レコードに対応するビットフィールドに１がセットされると、その論理レコードが他の論理レコードと依存関係の無いことを示す。一括転送中にエラーが発生した場合でも、この一括転送条件フィールドで１のセットされている論理レコードについては、受信処理を実行してもデータの到着順序で問題の生じることが無い。

【００４１】

また、後に明らかになるように、一括転送フラグの情報や、一括転送条件フィールドの情報をペイロードの中にいれて、あらかじめイニシエータからターゲットに通知しておいてもよい。その場合、各パケット中のトランスポートヘッダに、一括転送フラグや、一括転送条件フィールドがなくてもよい。

【００４２】

本実施例によれば、一括転送における一括転送条件をさらに詳細に規定することができる。したがって、一括転送中に転送エラーが生じた場合でも、データ依存関係がなく、到着順序で問題を生じる可能性の無い論理レコードについては、受信動作を継続することが可能となる。そのため、アプリケーションの処理効率を高めることができ、再送レコード数を減らすことでより効率のよい転送を実現することができる。

【００４３】

《実施例４》

本発明のデータ転送方法の動作フローを、図１０を用いて説明する。ここでは一括転送を開始するにあたり、イニシエータがターゲットに一括転送モード開始リクエストを発行している。これは、上述のように、一括転送条件フィールドなどの一括転送に関する情報を、通常のリライアブル送信パケットのペイロードに含めて送信するものである。ターゲットから当該パケットに対する受信完了通知が届いたら、イニシエータ・ターゲットとも一括転送モードになる。ここで、イニシエータからは、送信キューに格納されている一括転送リクエストの処理を順次開始し、ターゲットでは、それに対応したレスポンス処理を行う。一連の一括転送リクエストの処理が終了した時点で、ターゲットから一括転送モードの終了リクエストを、通常のリライアブル送信パケットとして送信する。

【００４４】

このとき、ターゲットでの一括転送受信記録である一括転送完了ステータスをペイロードの一部に含めて送信する。イニシエータからは、その受信完了通知を返信し、イニシエ

ータ・ターゲットは一括転送モードを解除する。イニシエータと、ターゲットは、一括転送完了ステータスを参照して、必要ならば、再送処理を開始する。すなわち、再送すべき論理レコードが存在すれば、それを順次、上記の手順に従い、一括転送する。一括転送リクエストのすべての論理レコードがターゲットに正常受信されたなら、一括転送を終了する。

【００４５】

本実施例において、ターゲットでは、転送エラーが発生した以降の当該一括転送モード中の論理レコード受信については無視し、アプリケーションの完了キューにも通知しない。そして、上記の一括転送完了ステータスの中に、一括転送モード中で転送エラーの検出された最初の論理レコードの転送ＩＤを含めて返送する。イニシエータでは、一括転送完了ステータスに含まれている、エラーの検出された論理レコードから再送を開始する。

【００４６】

本実施例に拠れば、一括転送中で転送エラーの生じるまでの正常受信が完了した論理レコードに関しては、受信完了後ただちに、対応するアプリケーションの処理を開始することができる。したがって、転送エラーが発生した際にも、アプリケーションの処理効率を高めることができ、再送レコード数を減らすことでより効率のよい転送を実現することができる。

【００４７】

《実施例５》

別の実施例では、上記の一括転送完了ステータスの中に、再送の必要な論理レコードの転送ＩＤリストを含めて返送する。イニシエータでは、ターゲットから通知された前記転送ＩＤリストを参照して再送処理を行う。

【００４８】

本実施例に拠れば、再送の必要な論理レコードについてのみ再送処理を行うことで、再送処理に要する時間を最小限に抑えることができる。したがって、再送レコード数を減らすことでより効率のよい転送を実現することができる。

【００４９】

《実施例６》

本発明のデータ転送方法における動作の詳細について、図１１、図１２を用いて説明を行う。図１１は、図１０内の一括転送モードにおけるＳＥＮＤリクエストの動作フローを示したものである。当該論理レコードが一括転送条件に合致していれば、送信処理１２０１を行い、正常に送信できたかどうか、送信ステータス記録処理１２２１を行う。図１２は、ＳＥＮＤリクエストに対応したレスポンスの動作フローを示している。当該論理レコードが一括転送条件に合致していれば、受信処理１１０１を行い、正常に受信できた場合完了キュー通知１１１０を行って、最後に受信ステータス記録処理１１２１を行う。

【００５０】

ＳＥＮＤリクエスト処理、ＳＥＮＤレスポンス処理とも、一括転送条件に合致しない論理レコードについては、当該レコードをスキップする。その際、相互にキャンセルリクエスト１１１２および１２１２を発行して、当該一括転送を途中で中止することが可能である。キャンセルリクエストを受けたほうは、その応答１１１１もしくは１２１１を返した後、一括転送モードを解除する。キャンセルリクエストを出したほうは、相手からその応答が戻ってきた後、一括転送モードを解除する。

【００５１】

本実施例に拠れば、一括転送の途中で転送エラー発生などにより、その後の一括転送の継続が不要になった場合、当該一括転送を途中で中止することが可能となる。そのため、不要な論理レコードの転送を抑制することで、より効率のよい転送を実現することができる。

【００５２】

《実施例７》

図１３、図１４、図１５に本発明のディスク制御装置に関する一実施例を示す。図１３

において、ディスク制御装置５００は、ホストコンピュータ５６０とホストインターフェースネットワーク５０１で接続している複数のホストインターフェース部５１０と、磁気ディスク装置５７０とディスクインターフェースネットワーク５０２で接続している複数のディスクインターフェース部５２０とを有しており、複数のホストインターフェース部５１０と複数のディスクインターフェース部５２０は、ディスク制御装置内ネットワーク５０３で接続されている。

【００５３】

図１４にホストインターフェース部５１０の構成を示す。複数のホストチャネルインターフェース５１１とプロセッサ５１２とメモリ５１３とＨＣＡ６０３を有し、それらがホストハブ５１４を介して接続されている。

【００５４】

図１５にディスクインターフェース部５２０の構成を示す。複数のディスクチャネルインターフェース５２１とプロセッサ５２２とメモリ５２３とＨＣＡ６０４を有し、それらがディスクハブ５２４を介して接続されている。また、ディスクハブ５２４には、キャッシュメモリ５２５も接続している。

【００５５】

各ホストインターフェース部５１０は、ホストコンピュータ５６０とのインターフェースとキャッシュメモリ５２５との間のデータ転送を実行し、各ディスクインターフェース部５２０は、磁気ディスク装置５７０とのインターフェースとキャッシュメモリ５２５との間のデータ転送を実行する。

【００５６】

ホストインターフェース部５１０と、ディスクインターフェース部５２０は、ＨＣＡ６０３とＨＣＡ６０４を介してデータ転送を行う。その際、ホストインターフェース部は、複数のホストチャネルからのコマンドやデータをホストハブでまとめて、ディスクインターフェース部に転送する。その際に、上述のデータ転送方法を用いる。なお、ＨＣＡは、図５に示した機能と同等の機能を有するものであればよく、例えば、その一部の機能をプロセッサ５１２やプロセッサ５２２のソフトウェア処理により実現してもかまわない。

【００５７】

本実施例によると、ホストコンピュータ部とディスクインターフェース部の転送効率を大幅に改善することができる。特に、ディスク制御装置内ネットワーク５０３の転送効率の向上と、ホストインターフェース部、および、ディスクインターフェース部からのＩＯリクエストに対するレスポンス時間の短縮が可能となり、ディスク制御装置の処理性能を高めることができる。

【００５８】

《実施例８》

図１３に示したディスク制御装置は、その信頼性を向上させるため、複数のディスクインターフェース部に内蔵されているキャッシュメモリに、冗長にデータを格納している。そのため、ホストインターフェース部５１０から、ディスクインターフェース部５２０に書き込み要求があった場合、あるディスクインターフェース部内のキャッシュメモリ５２５にデータを格納した後、別のディスクインターフェース部内のキャッシュメモリに対しても、冗長書き込みを実行する。この複数のディスクインターフェース部間での冗長書き込みに、上述のデータ転送方法を使用する。

【００５９】

冗長書き込みは、ディスク制御装置の信頼性向上のために不可欠ではあるが、それによるディスク制御装置内ネットワークへの負荷増加とそれに起因するシステム性能劣化が課題であった。本実施例により、ディスク制御装置内ネットワークの転送効率を向上させて、ディスク制御装置の処理性能を高めることが可能となる。

【００６０】

《実施例９》

図１３に示したディスク制御装置では、仮想化機能実現などのために他のディスク制御

装置と連携して動作する場合に、ホストインターフェース部間でデータ転送を行う必要がある。このホストインターフェース部間のデータ転送に、上述のデータ転送方法を適用する。

【００６１】

ホストインターフェース部間転送の場合、複数のディスク制御装置を経由してのデータアクセスとなるので、そのレスポンス時間を、可能な限り短くする必要がある。本実施例によるとホストインターフェース部間でのレスポンス時間の大幅な短縮が可能となり、ディスク制御装置の処理性能が向上する。

【００６２】

《実施例１０》

図１６に示したディスク制御装置では、ディスク制御装置５００が、ホストコンピュータ５６０とホストインターフェースネットワーク５０１で接続している複数のホストインターフェース部６１０と、磁気ディスク装置５７０とディスクインターフェースネットワーク５０２で接続している複数のディスクインターフェース部６２０と、複数のメモリ部５８０と、複数のプロッセサ部５９０を有しており、複数のホストインターフェース部６１０と複数のディスクインターフェース部６２０と、複数のメモリ部５８０と、複数のプロセッサ部５９０とは、ディスク制御装置内ネットワーク５０３で接続されている。

【００６３】

前記複数のホストインターフェース部５０１は、上記複数のプロセッサ５９０と連携して上記メモリ部５８０とデータ転送を実行し、あるいは複数のディスクインターフェース部６２０は上記プロセッサ部５９０と連携して上記磁気ディスク装置５７０と上記メモリ部５８０との間でデータ転送を実行する。このデータ転送に上述の本発明のデータ転送方法を用いる。

【００６４】

本実施例によると、ディスク制御装置内ネットワーク５０３の転送効率の向上と、ホストインターフェース部５０１、および、ディスクインターフェース部６２０からのＩ／Ｏリクエストに対するレスポンス時間の短縮が可能となり、ディスク制御装置の処理性能を高めることができる。

【００６５】

また、上記複数のメモリ部５８０とプロッセサ部５９０間のデータ転送や上記複数のメモリ部５８０間、あるいは、プロッセサ部５９０間のデータ転送にも上述の本発明の高速データ転送方法を用いることができ、同様にディスク制御装置の処理性能を高めることができる。

【図面の簡単な説明】

【００６６】

【図１】本発明によるデータ転送方法の動作原理を示す図である。

【図２】従来のデータ転送方法の動作原理を示す図である。

【図３】従来のデータ転送方法の動作原理を示す図である。

【図４】IOシステムの構成を示す図である。

【図５】ホストチャネルアダプタ（ＨＣＡ）の構成を示す図である。

【図６】従来の単純データ転送方法の動作詳細を示す図である。

【図７】従来のRDMAデータ転送方法の動作詳細を示す図である。

【図８】本発明のデータ転送方法で使用する転送リクエストと論理レコードの構造を示す図である。

【図９】本発明のデータ転送方法で使用するパケット構造を示す図である。

【図１０】本発明のデータ転送方法の動作フローを示す図である。

【図１１】本発明のデータ転送方法におけるSENDリクエストの動作フローを示す図である。

【図１２】本発明のデータ転送方法におけるSENDレスポンスの動作フローを示す図である。

【図１３】本発明のディスク制御装置の構成を示す図である。
【図１４】本発明のディスク制御装置で使用されるホストインターフェース部の構成を示す図である。
【図１５】本発明のディスク制御装置で使用されるディスクインターフェース部の構成を示す図である。
【図１６】本発明のディスク制御装置の構成を示す図である。
【図１７】本発明のディスク制御装置で使用されるホストインターフェース部の構成を示す図である。
【図１８】本発明のディスク制御装置で使用されるディスクインターフェース部の構成を示す図である。

【符号の説明】

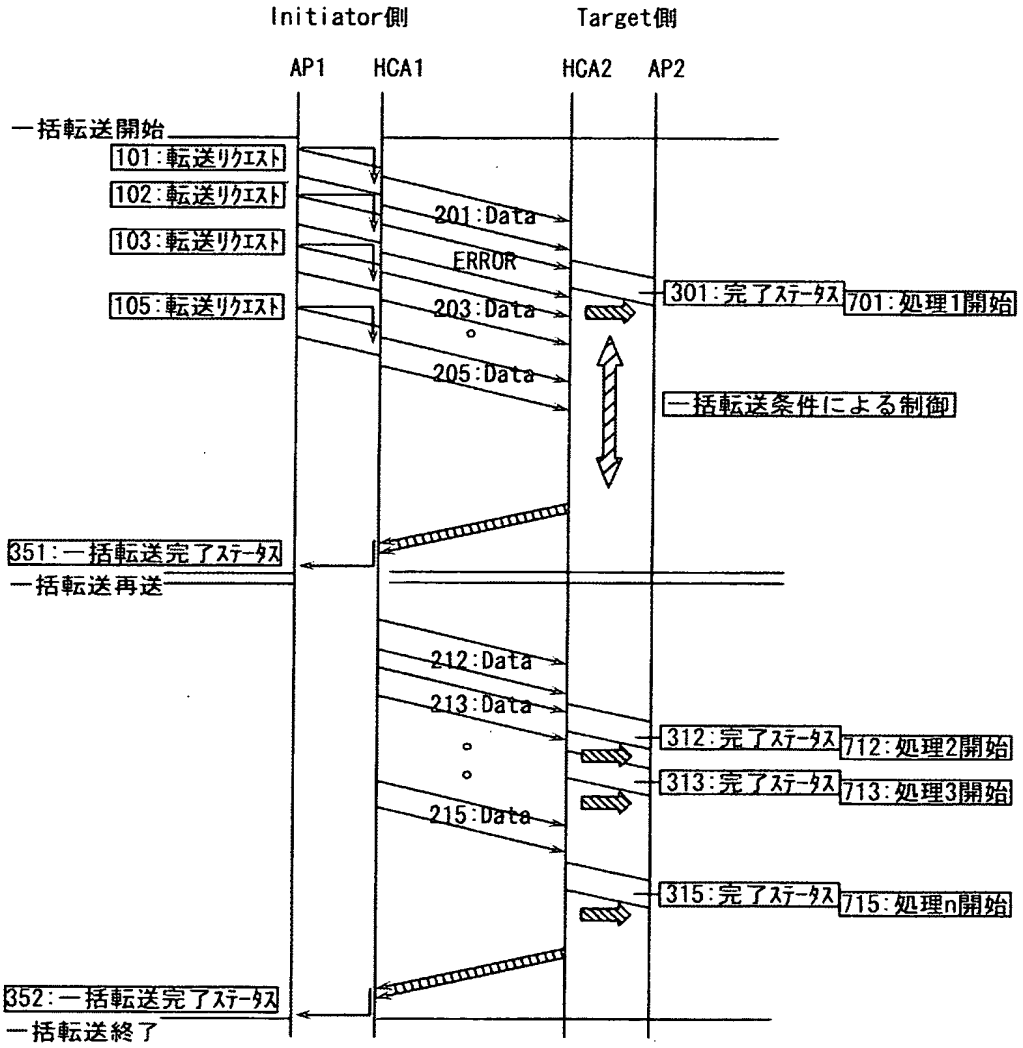
【００６７】

１１、１２、１３、１４、１５、１６、１７、１８、１９…送信キュー、
２１、２２、２３、２４、２５、２６、２７、２８…受信キュー、
３１、３２、３３、３４、３５、３６、３９…完了キュー、
４１、４２、４３、４４、４５、４６、４７、４８…キューペア、
５１、５２、５３、５４…プロセス、
６１、６２…プロセスコード、
７１、７２、７３、７４…プロセスバッファ、
７５、７６…アプリケーションメモリ空間、
８１、８２、８３、８４、８９…レコードバッファ、
８５、８６、８７…ＲＤＭＡ転送空間、
１００、１０１、１０２、１０３、１０５、１２１、１２５、１３１…転送リクエスト、
２０１、２０５、２１２、２１３、２１５、２２１、２２５、２３１、２３５、２３２、２３４…転送データ、
３００、３０１、３１２、３１３、３１５、３２１、３２５、３３１…(論理レコード)完了ステータス、
３５１、３５２…一括転送完了ステータス、
３６１、３６５、３７１…転送完了ステータス、
４０１、４０２、４０３、４０４、４１１、４１３、４１４、４２１、４２２、４２３、４２４、４３１、４３３、４３４、４４０…パケット、
４４１…ルーティングヘッダ、
４４２…トランスポートヘッダ
４４３…ペイロード、
４４４…ＣＲＣ、
４５０…一括転送フラグ、
４５１、４５６…一括転送条件フィールド、
４５５…一括転送レコード数、
５００…ディスク制御装置、
５０１…ホストインターフェースネットワーク、
５０２…ディスクインターフェースネットワーク、
５０３…ディスク制御装置内ネットワーク、
５１０…ホストインターフェース部、
５１１…ホストチャネルインターフェース、
５１２…プロセッサ、
５１３…メモリ、
５１４…ホストハブ、
５２０…ディスクインターフェース部、
５２１…ディスクチャネルインターフェース、

５２２…プロセッサ、
５２３…メモリ、
５２４…ディスクハブ、
５２５…キャッシュメモリ
５６０…ホストコンピュータ、
５７０…磁気ディスク装置、
５８０…メモリ、
５９０…プロセッサ、
６０１、６０２、６０３、６０４…ホストチャネルアダプタ、
６１０…ホストインターフェース部、
６１１、６１２、６１３…受信ポート、
６２０…プロセッサー部、
６２１、６２２、６２３…送信ポート、
６３１…受信リンク論理、
６３２…受信トランスポート論理、
６３３…受信プロセッサ、
６４１…送信リンク論理、
６４２…送信トランスポート論理、
６４３…送信プロセッサ、
６５０…メモリ、
６６０…接続インターフェース、
６７０…内部バス
７０１、７１２、７１３、７１５、７２１、７２５、７３１…処理開始、
１００１…一括送信モード開始リクエスト、
１００２…一括転送モード開始レスポンス、
１０１１…送信リクエスト、
１０１３…ＲＤＭＡ送信リクエスト、
１０１５…送信レスポンス、
１０１７…ＲＤＭＡ送信レスポンス、
１０２１…一括転送モード終了レスポンス、
１０２２…一括転送モード終了リクエスト
１１０１…受信、
１１１０…完了通知、
１１１１…キャンセルレスポンス、
１１１２…キャンセルリクエスト、
１１２１…受信ステータス記録、
１２０１…送信、
１２１１…キャンセルレスポンス、
１２１２…キャンセルリクエスト、
１２２１…送信ステータス記録。

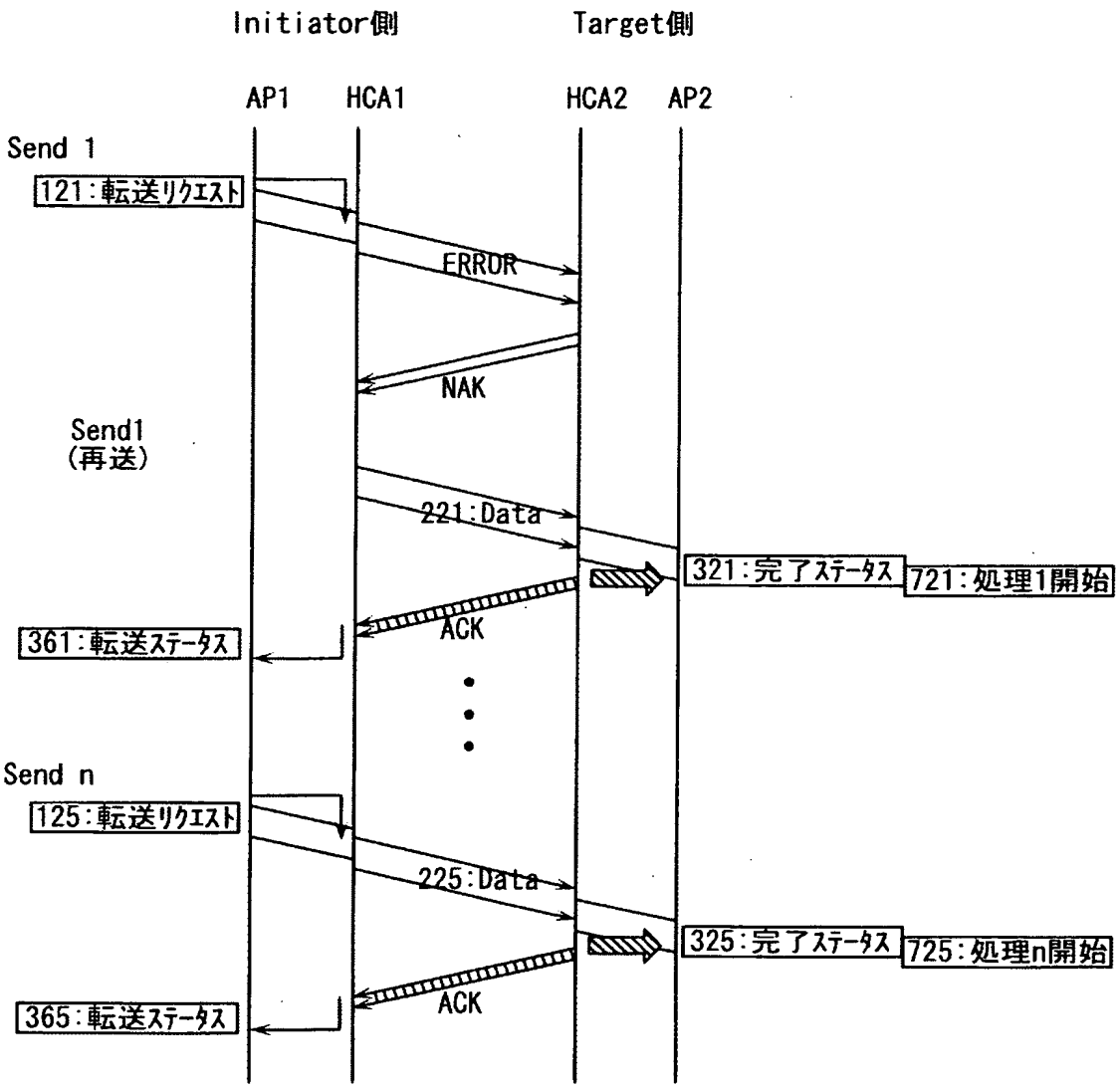
【書類名】図面
【図１】

図　1


Initiator側
Target側
AP1
HCA1
HCA2
AP2
一括転送開始
101:転送リクエスト
102:転送リクエスト
103:転送リクエスト
105:転送リクエスト
201:Data
ERROR
203:Data
205:Data
301:完了ステータス
701:処理1開始
一括転送条件による制御
351:一括転送完了ステータス
一括転送再送
212:Data
213:Data
215:Data
312:完了ステータス
712:処理2開始
313:完了ステータス
713:処理3開始
315:完了ステータス
715:処理n開始
352:一括転送完了ステータス
一括転送終了

【図２】

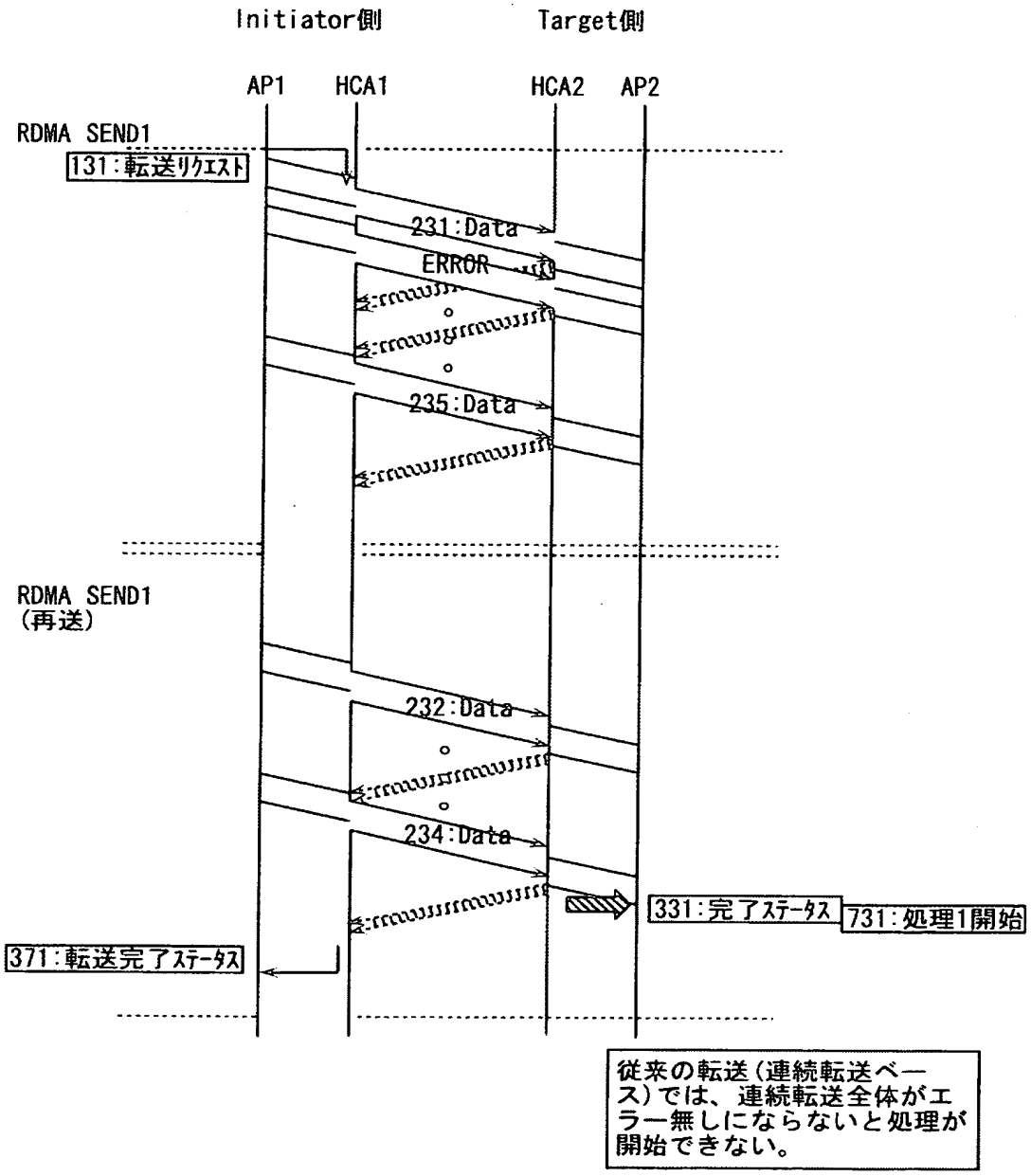
図　2



従来の転送（単純転送ベース）では、データ転送前後のＯＨが転送ごとに発生するため、転送効率を上げられない。

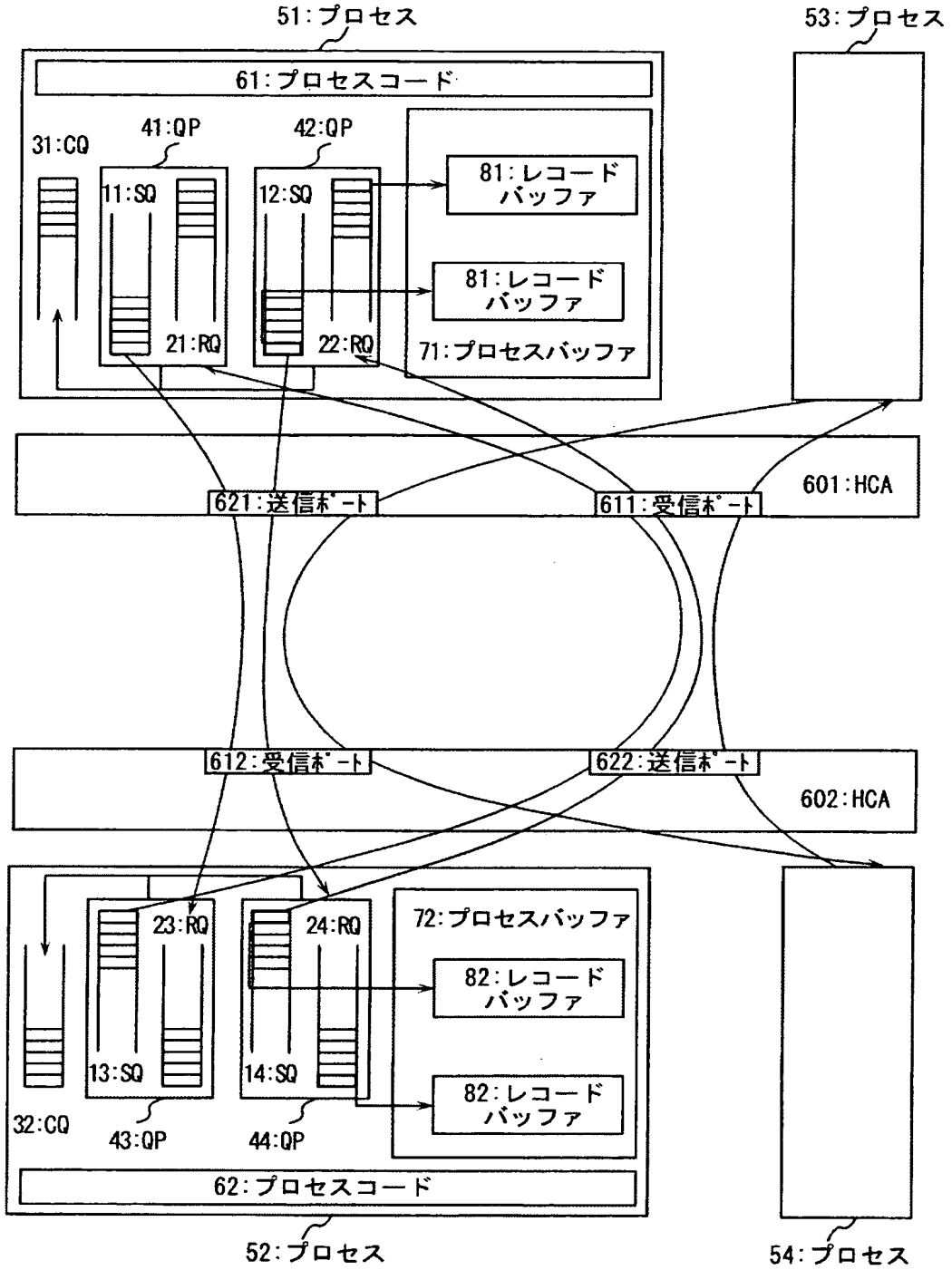
【図３】

図　３


Initiator側
Target側
AP1
HCA1
HCA2
AP2
RDMA SEND1
131:転送リクエスト
231:Data
ERROR
235:Data
RDMA SEND1
(再送)
232:Data
234:Data
331:完了ステータス
731:処理1開始
371:転送完了ステータス
従来の転送(連続転送ベース)では、連続転送全体がエラー無しにならないと処理が開始できない。

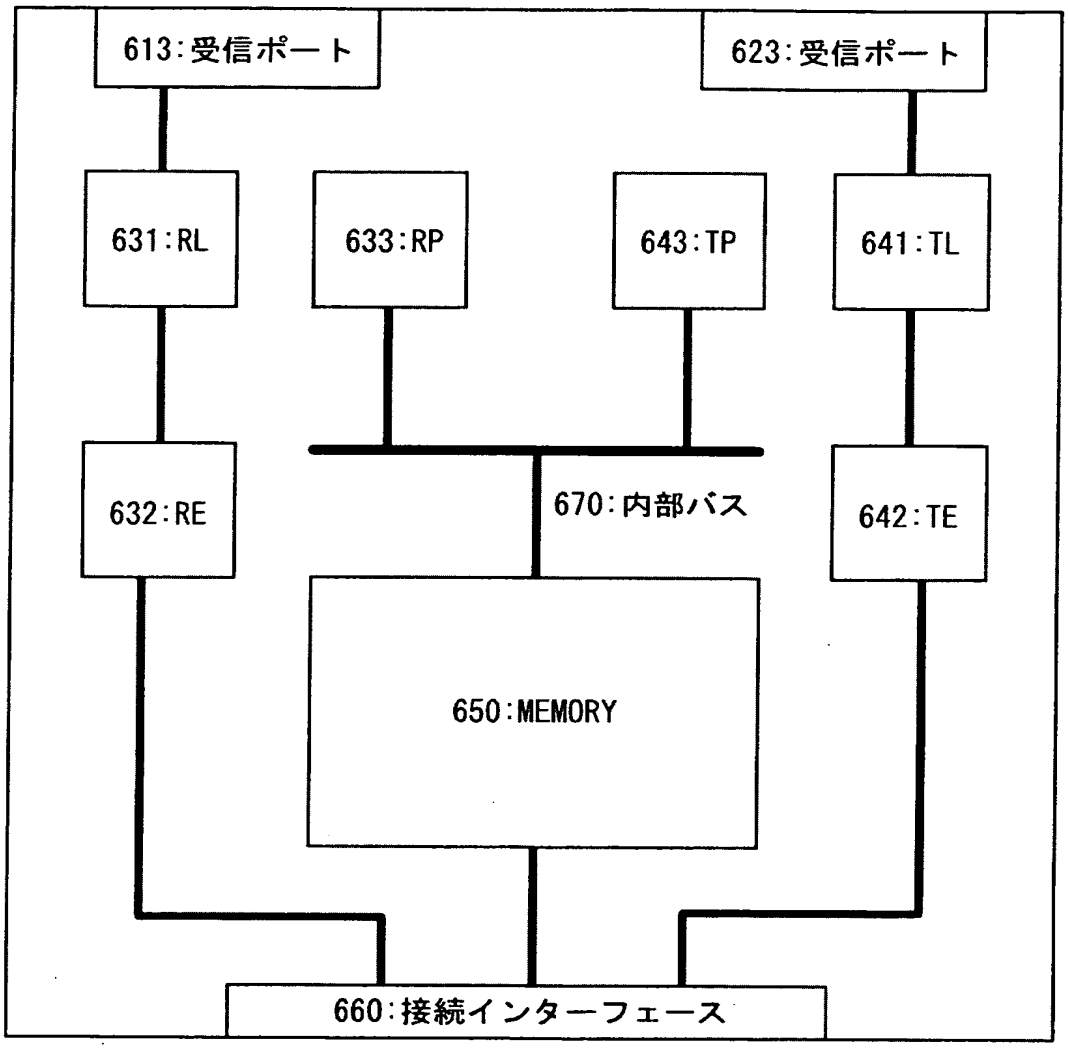
【図４】

図　4


51:プロセス
53:プロセス
61:プロセスコード
31:CQ
41:QP
42:QP
11:SQ
12:SQ
81:レコード
バッファ
81:レコード
バッファ
21:RQ
22:RQ
71:プロセスバッファ
601:HCA
621:送信ポート
611:受信ポート
612:受信ポート
622:送信ポート
602:HCA
23:RQ
24:RQ
72:プロセスバッファ
82:レコード
バッファ
82:レコード
バッファ
13:SQ
14:SQ
32:CQ
43:QP
44:QP
62:プロセスコード
52:プロセス
54:プロセス

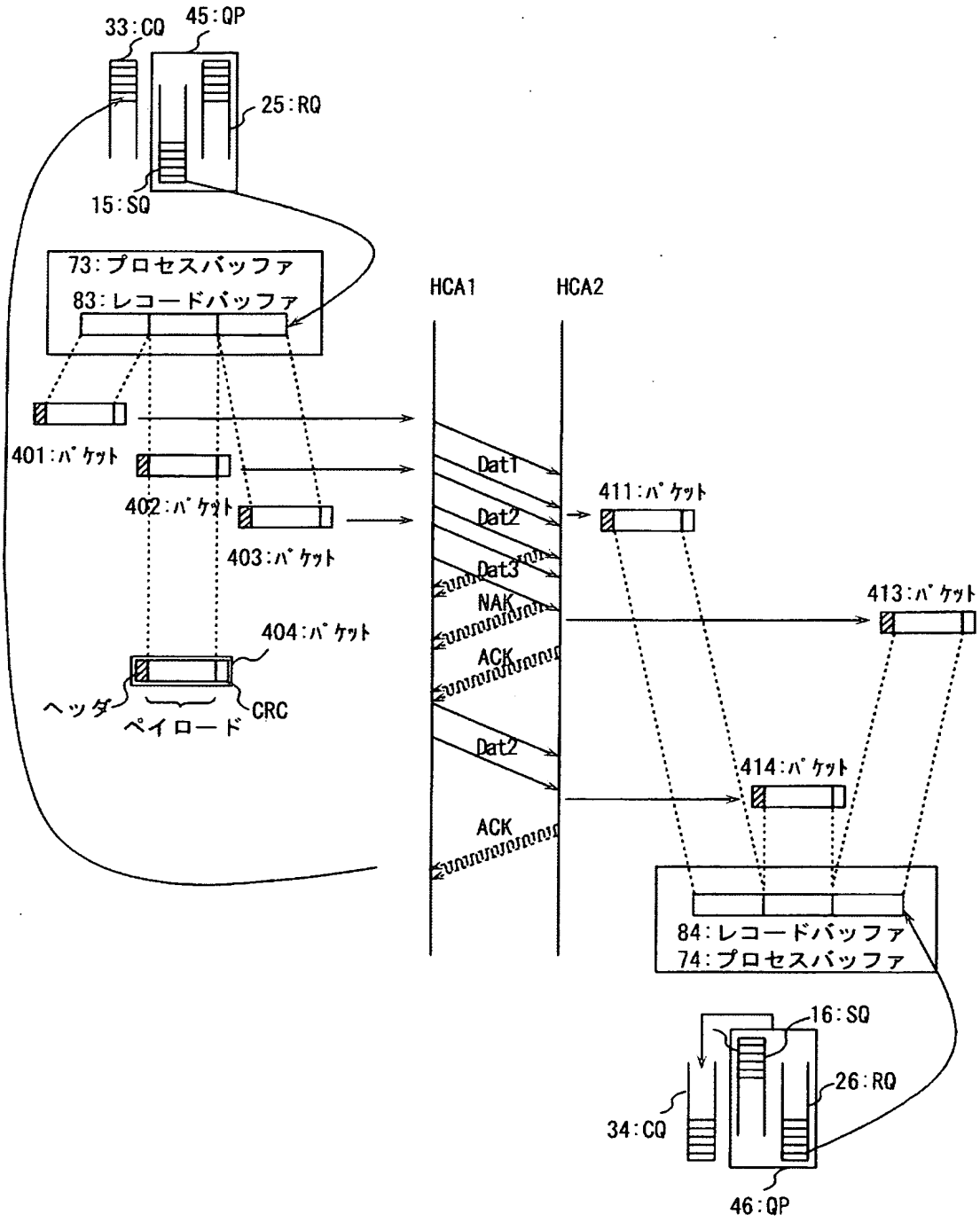
【図５】

図　５


613:受信ポート
623:受信ポート
631:RL
633:RP
643:TP
641:TL
632:RE
670:内部バス
642:TE
650:MEMORY
660:接続インターフェース

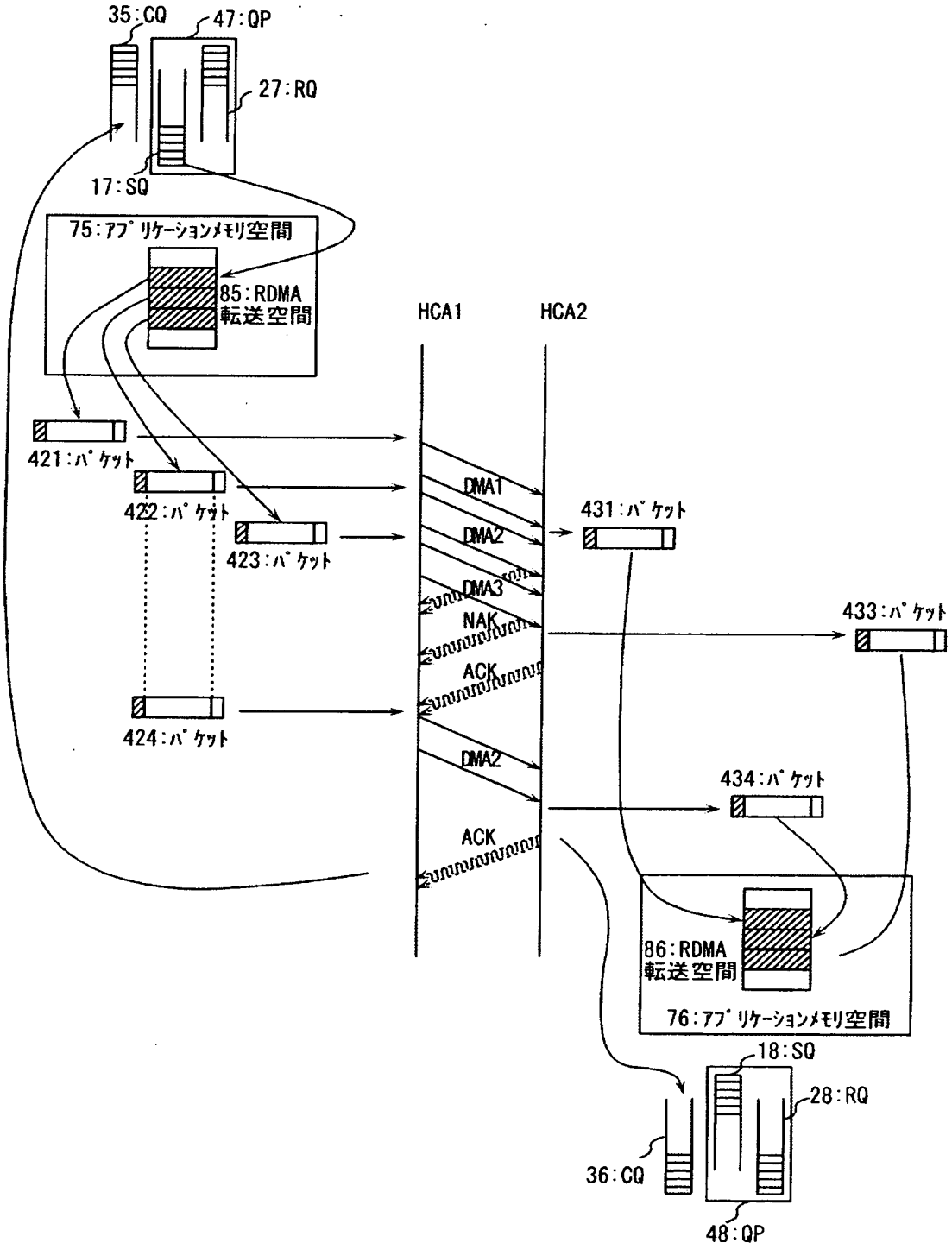
【図６】

図　6


33:CQ
45:QP
25:RQ
15:SQ
73:プロセスバッファ
83:レコードバッファ
HCA1
HCA2
401:パケット
402:パケット
403:パケット
404:パケット
ヘッダ
ペイロード
CRC
Dat1
Dat2
Dat3
NAK
ACK
411:パケット
413:パケット
414:パケット
84:レコードバッファ
74:プロセスバッファ
16:SQ
26:RQ
34:CQ
46:QP

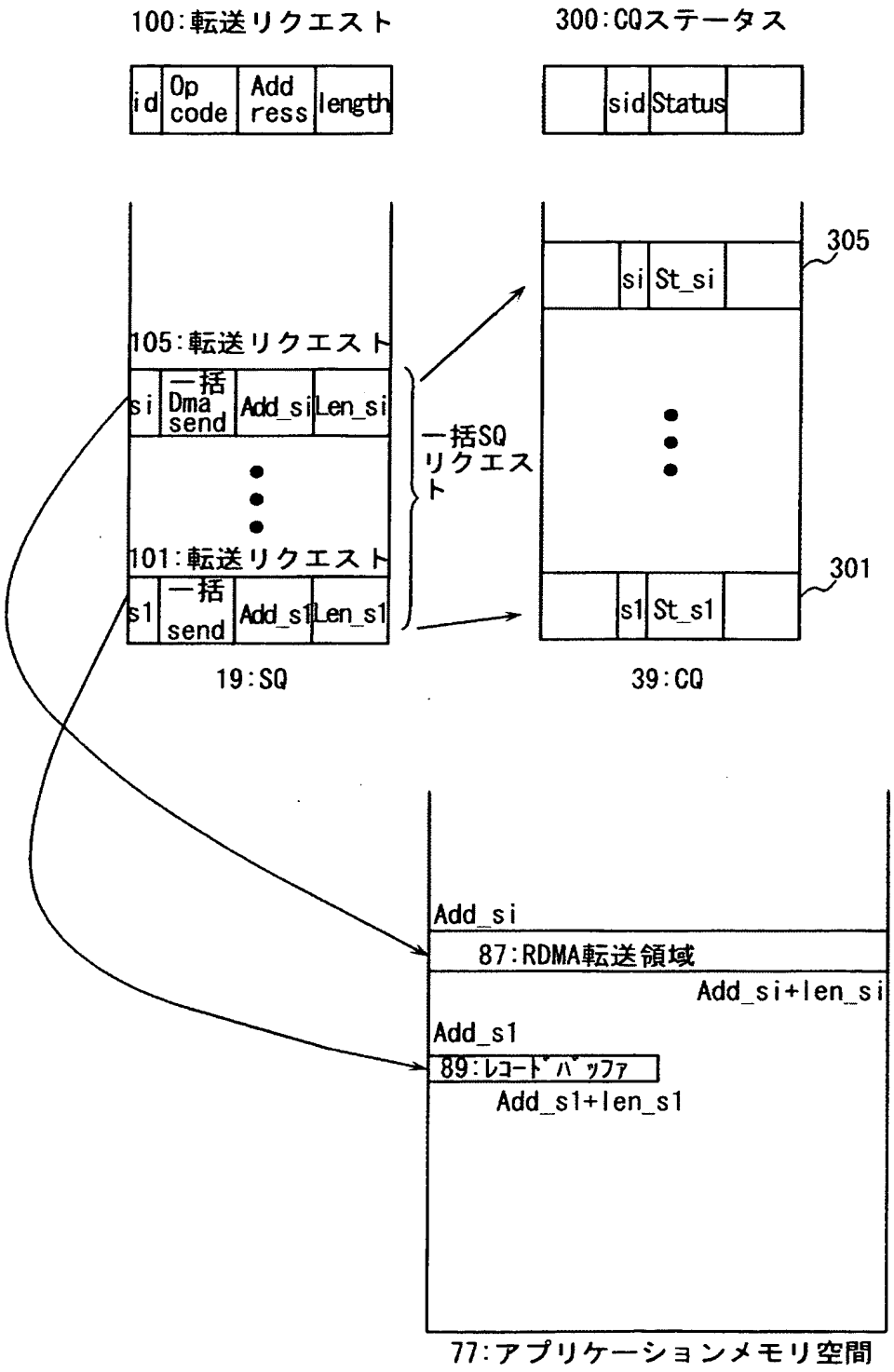
【図７】

図　7


35:CQ
47:QP
27:RQ
17:SQ
75:アプリケーションメモリ空間
85:RDMA
転送空間
HCA1
HCA2
421:パケット
422:パケット
423:パケット
424:パケット
DMA1
DMA2
DMA3
NAK
ACK
DMA2
ACK
431:パケット
433:パケット
434:パケット
86:RDMA
転送空間
76:アプリケーションメモリ空間
18:SQ
28:RQ
36:CQ
48:QP

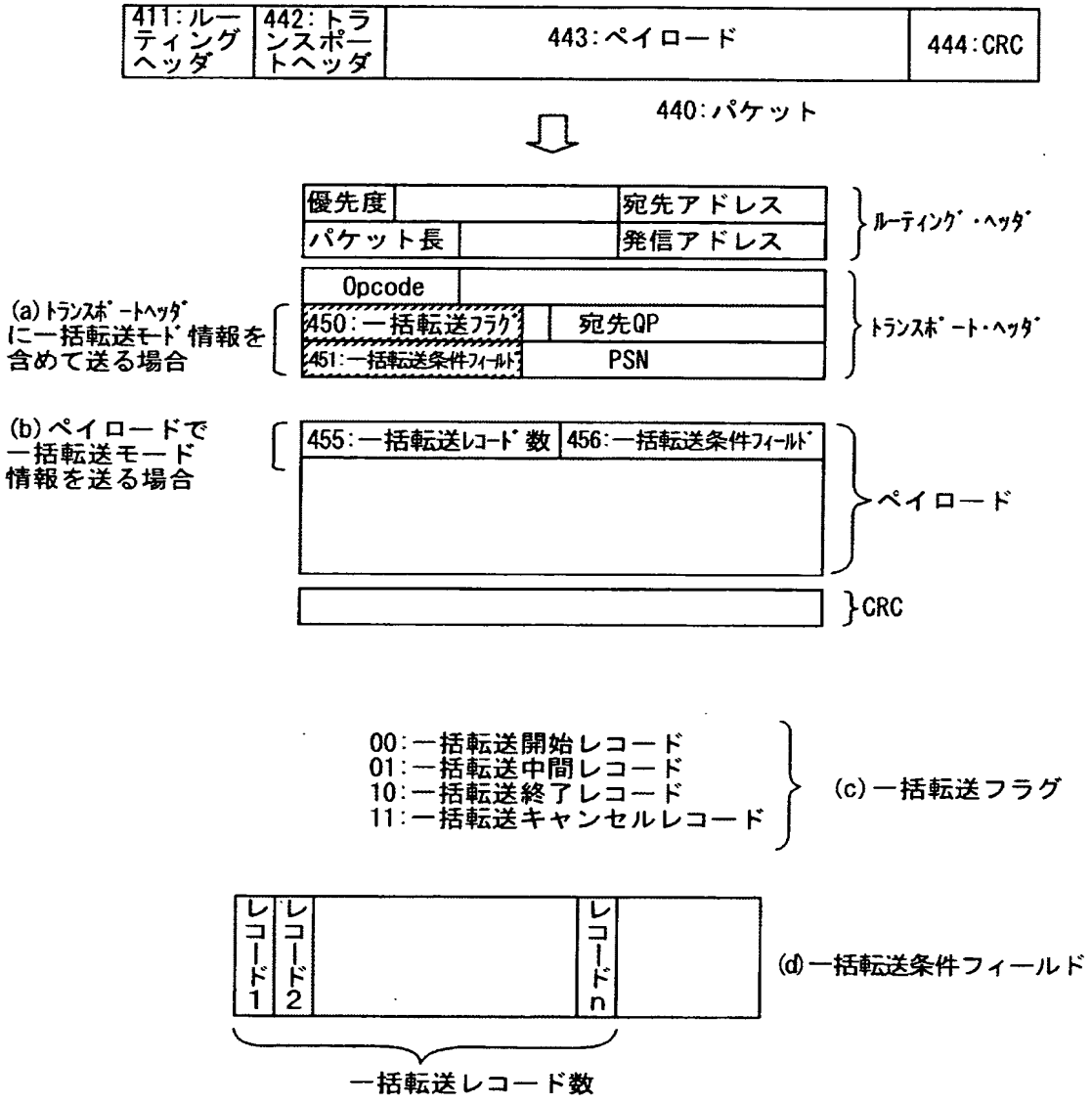
【図８】

図　8


100:転送リクエスト
id
Op code
Add ress
length
300:CQステータス
sid
Status
305
si
St_si
105:転送リクエスト
si
一括 Dma send
Add_si
Len_si
一括SQ リクエスト
101:転送リクエスト
s1
一括 send
Add_s1
Len_s1
301
s1
St_s1
19:SQ
39:CQ
Add_si
87:RDMA転送領域
Add_si+len_si
Add_s1
89:ﾚｺｰﾄﾞﾊﾞｯﾌｧ
Add_s1+len_s1
77:アプリケーションメモリ空間

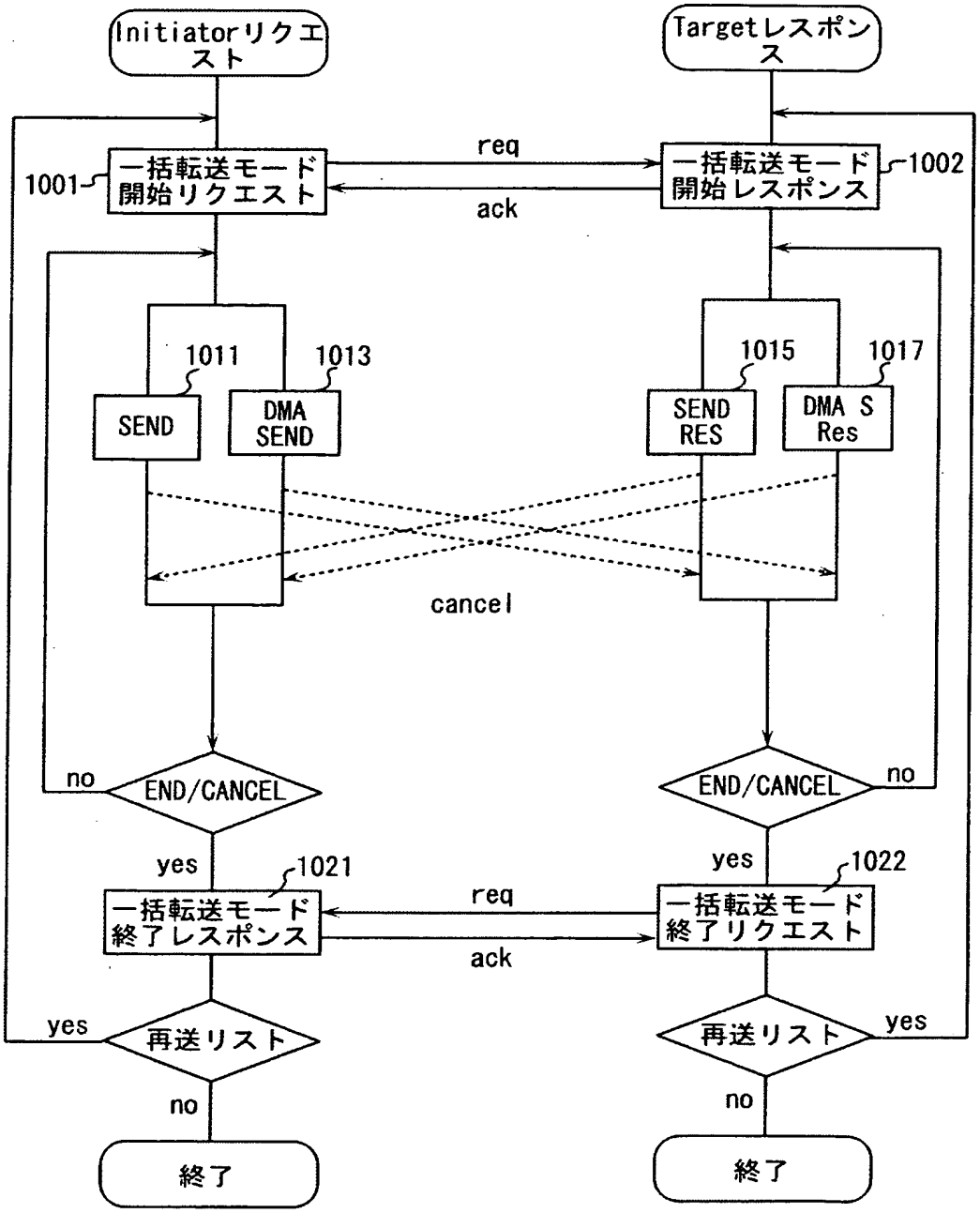
【図９】

図　９


411:ルーティングヘッダ
442:トランスポートヘッダ
443:ペイロード
444:CRC
440:パケット
優先度
宛先アドレス
パケット長
発信アドレス
ルーティング・ヘッダ
Opcode
450:一括転送フラグ
宛先QP
451:一括転送条件フィールド
PSN
トランスポート・ヘッダ
(a)トランスポートヘッダに一括転送モード情報を含めて送る場合
(b)ペイロードで一括転送モード情報を送る場合
455:一括転送レコード数
456:一括転送条件フィールド
ペイロード
CRC
00:一括転送開始レコード
01:一括転送中間レコード
10:一括転送終了レコード
11:一括転送キャンセルレコード
(c)一括転送フラグ
レコード1
レコード2
レコードn
(d)一括転送条件フィールド
一括転送レコード数

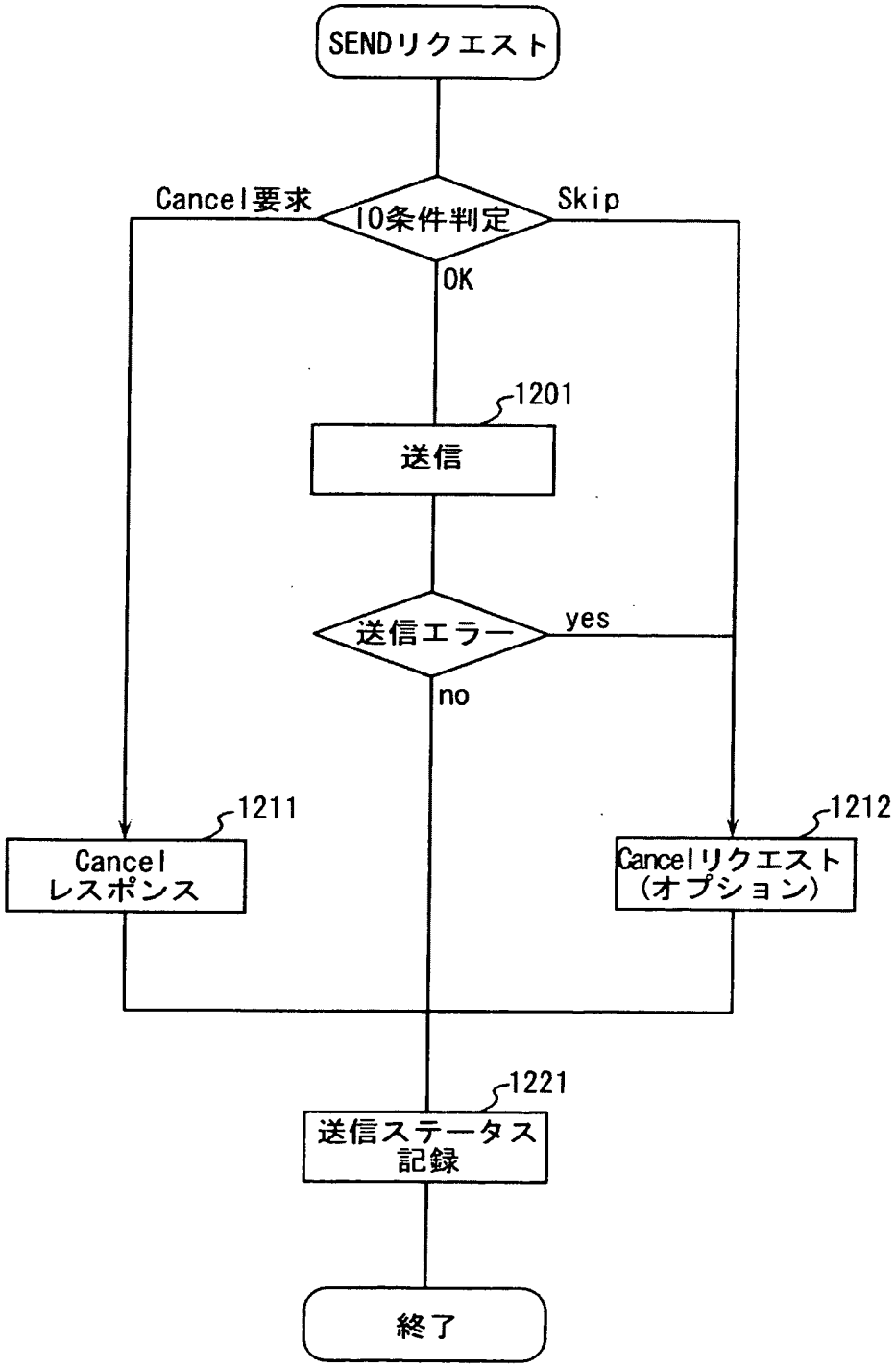
【図１０】

図　１０


Initiatorリクエスト
Targetレスポンス
一括転送モード開始リクエスト
1001
req
ack
一括転送モード開始レスポンス
1002
1011
SEND
1013
DMA SEND
1015
SEND RES
1017
DMA S Res
cancel
no
END/CANCEL
yes
END/CANCEL
no
yes
1021
一括転送モード終了レスポンス
req
ack
1022
一括転送モード終了リクエスト
yes
再送リスト
no
終了
再送リスト
yes
no
終了

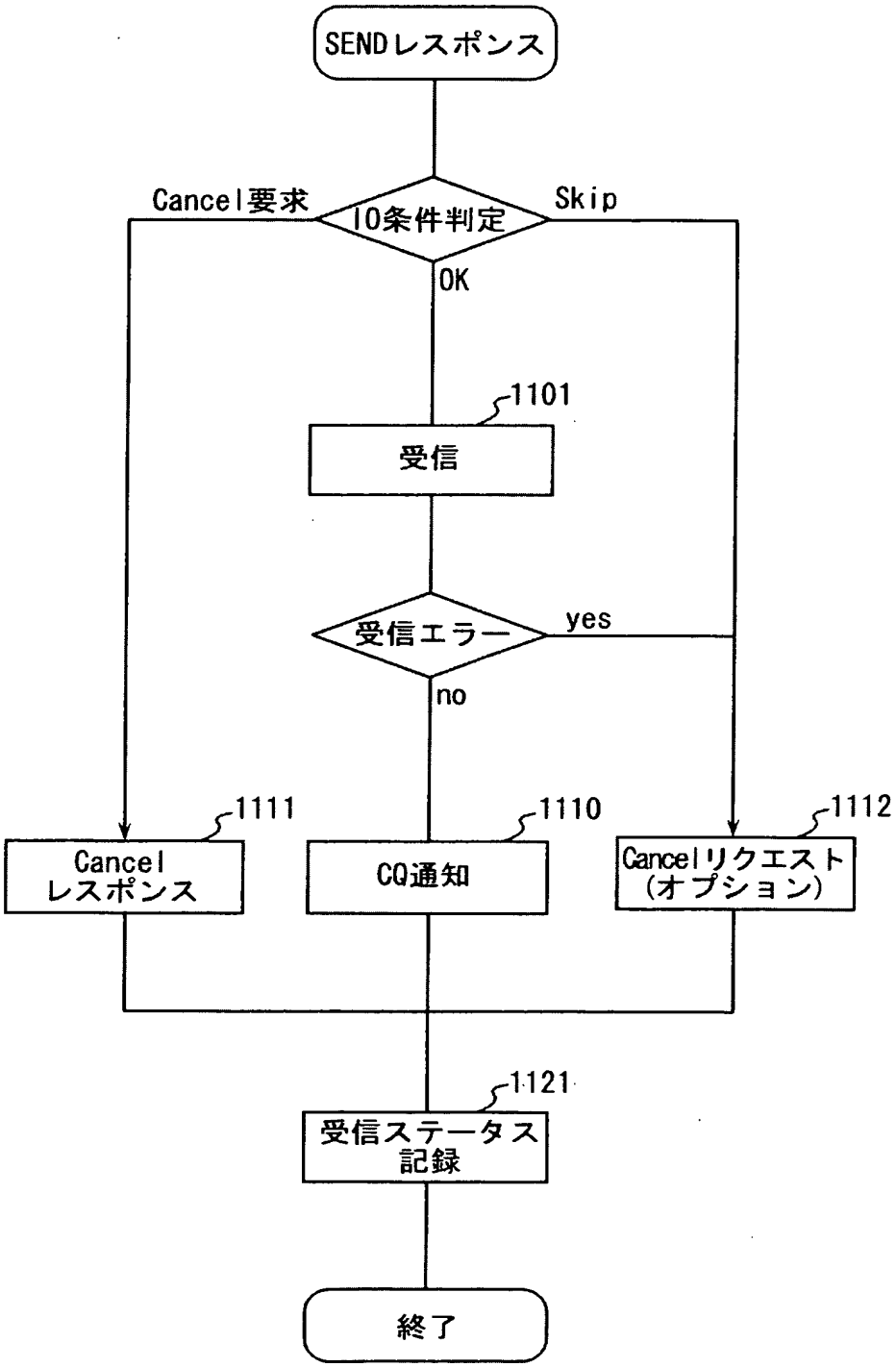
【図１１】

図　１１


SENDリクエスト
Cancel要求
IO条件判定
Skip
OK
1201
送信
送信エラー
yes
no
1211
Cancel
レスポンス
1212
Cancelリクエスト
(オプション)
1221
送信ステータス
記録
終了

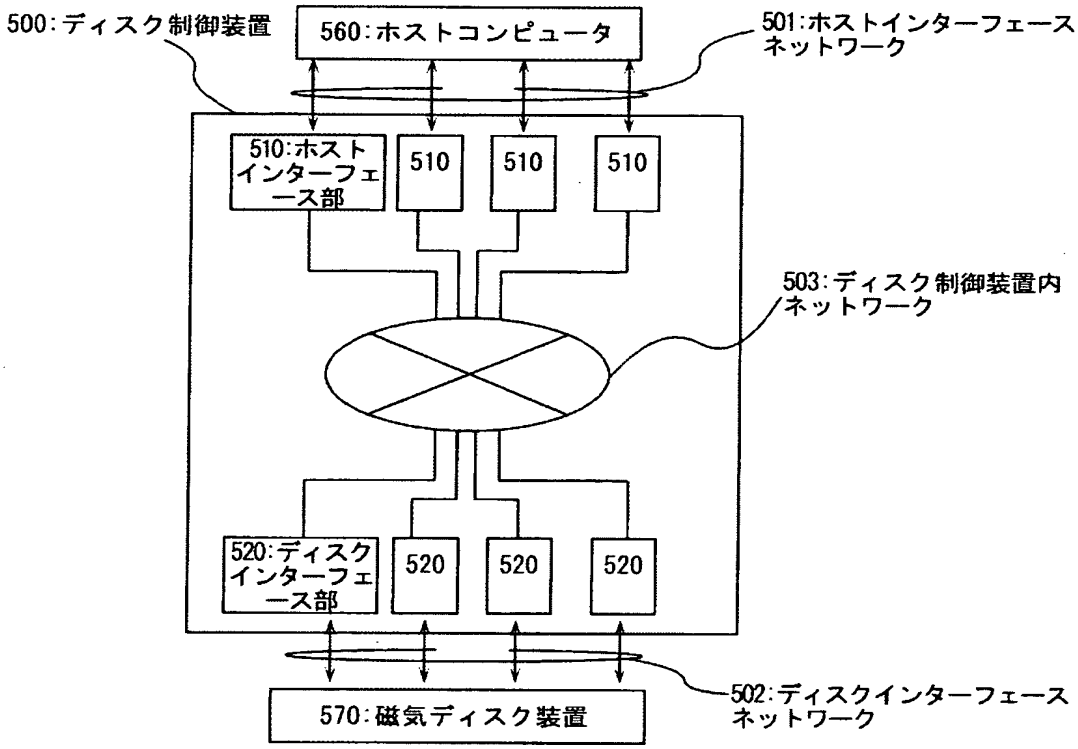
【図１２】

図　１２


SENDレスポンス
Cancel要求
IO条件判定
Skip
OK
1101
受信
受信エラー
yes
no
1111
1110
1112
Cancel
レスポンス
CQ通知
Cancelリクエスト
(オプション)
1121
受信ステータス
記録
終了

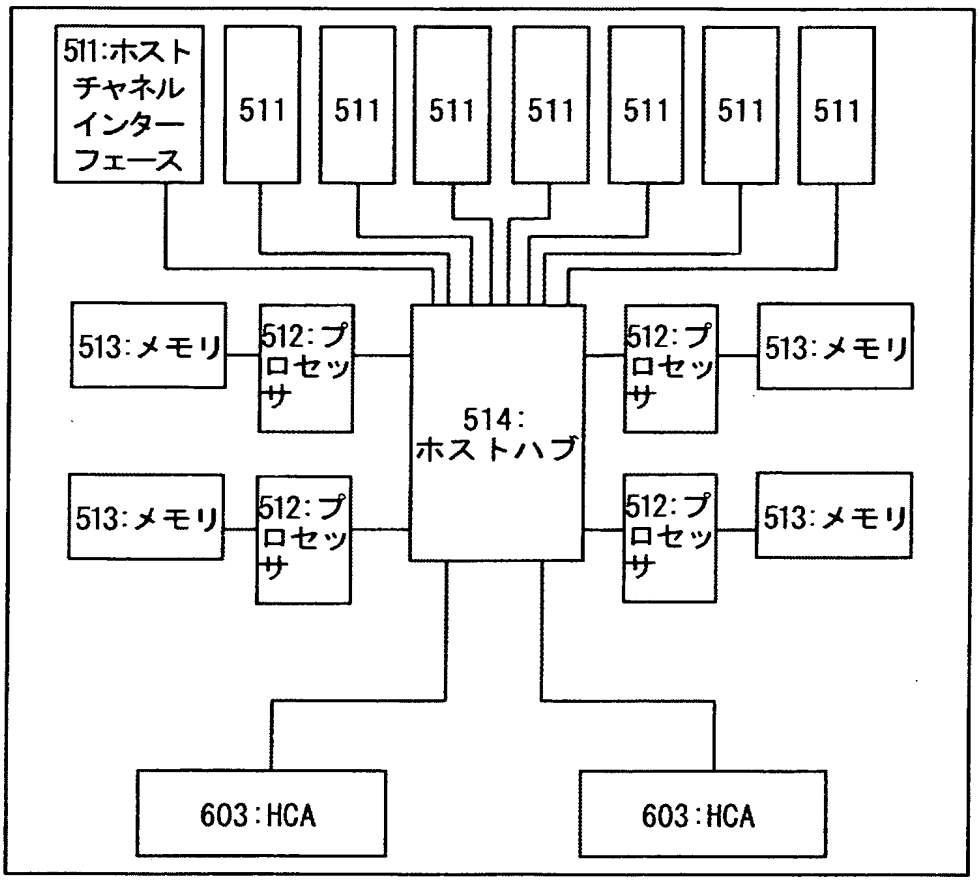
【図１３】

図　１３


500:ディスク制御装置
560:ホストコンピュータ
501:ホストインターフェース
ネットワーク
510:ホスト
インターフェ
ース部
510
510
510
503:ディスク制御装置内
ネットワーク
520:ディスク
インターフェ
ース部
520
520
520
502:ディスクインターフェース
ネットワーク
570:磁気ディスク装置

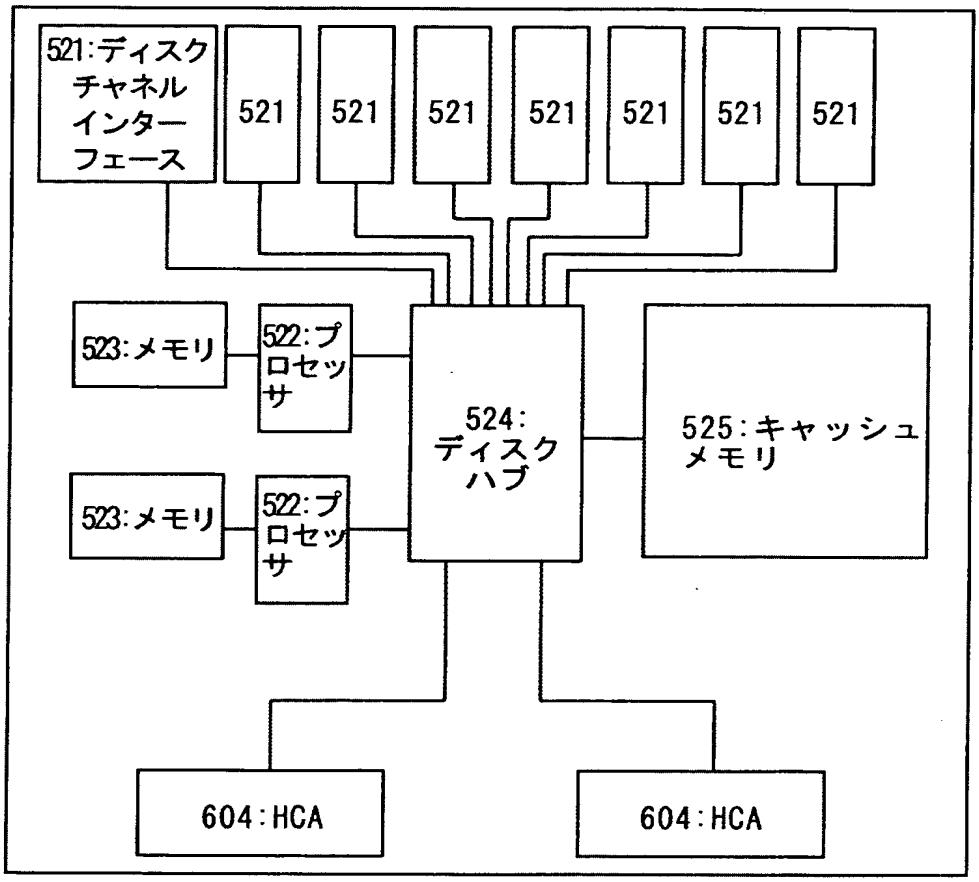
【図１４】

図　１４


511:ホストチャネルインターフェース
511
511
511
511
511
511
511
513:メモリ
512:プロセッサ
514:ホストハブ
512:プロセッサ
513:メモリ
513:メモリ
512:プロセッサ
512:プロセッサ
513:メモリ
603:HCA
603:HCA

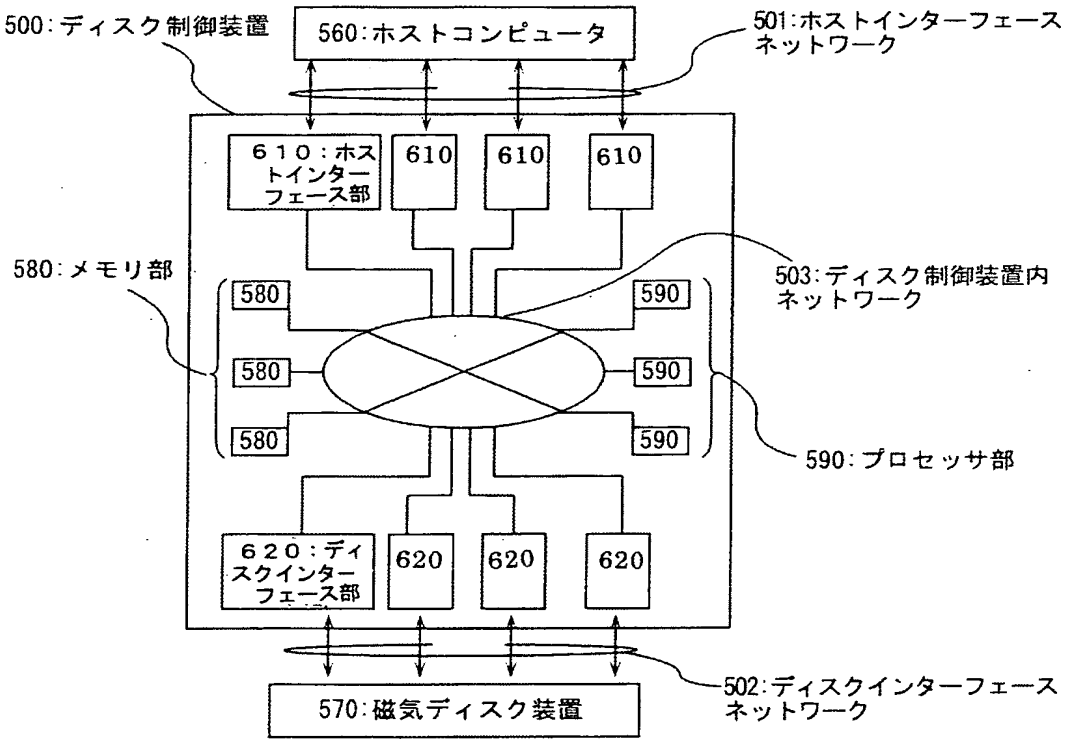
【図１５】

図　１５


521:ディスクチャネルインターフェース
521
521
521
521
521
521
521
523:メモリ
522:プロセッサ
524:ディスクハブ
525:キャッシュメモリ
523:メモリ
522:プロセッサ
604:HCA
604:HCA

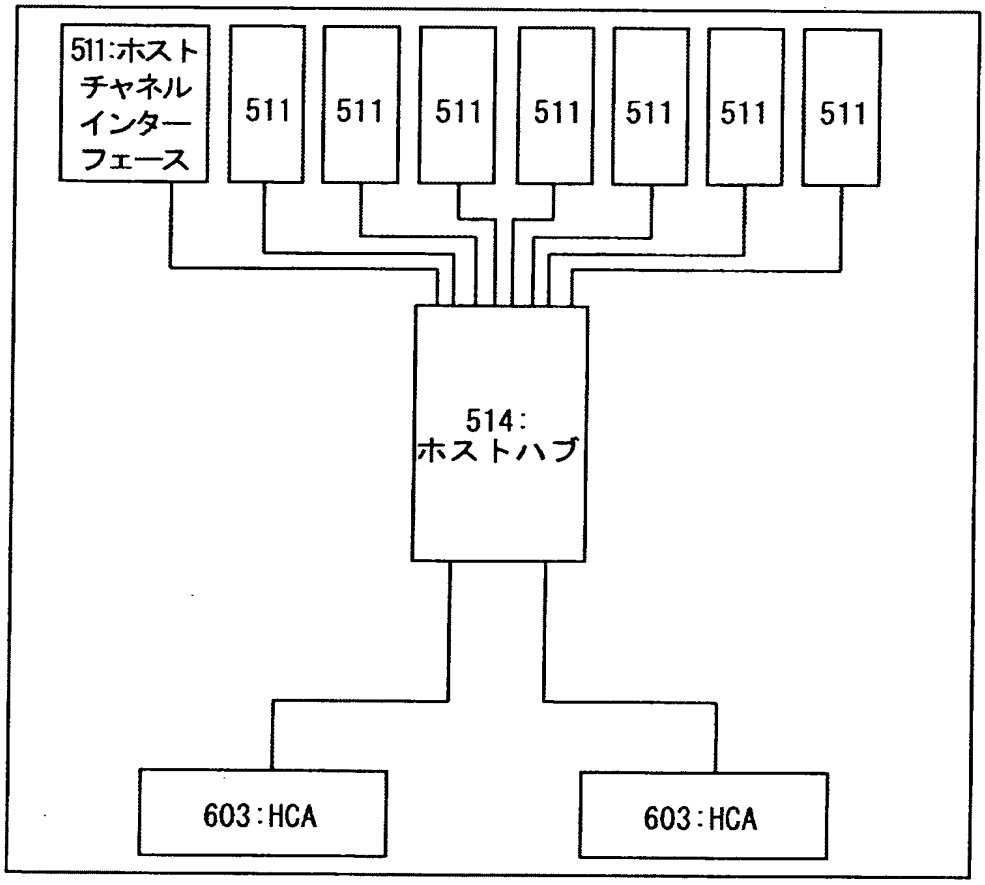
【図１６】

図　１６


500:ディスク制御装置
560:ホストコンピュータ
501:ホストインターフェース
ネットワーク
６１０:ホス
トインター
フェース部
610
610
610
580:メモリ部
503:ディスク制御装置内
ネットワーク
580
580
580
590
590
590
590:プロセッサ部
６２０:ディ
スクインター
フェース部
620
620
620
502:ディスクインターフェース
ネットワーク
570:磁気ディスク装置

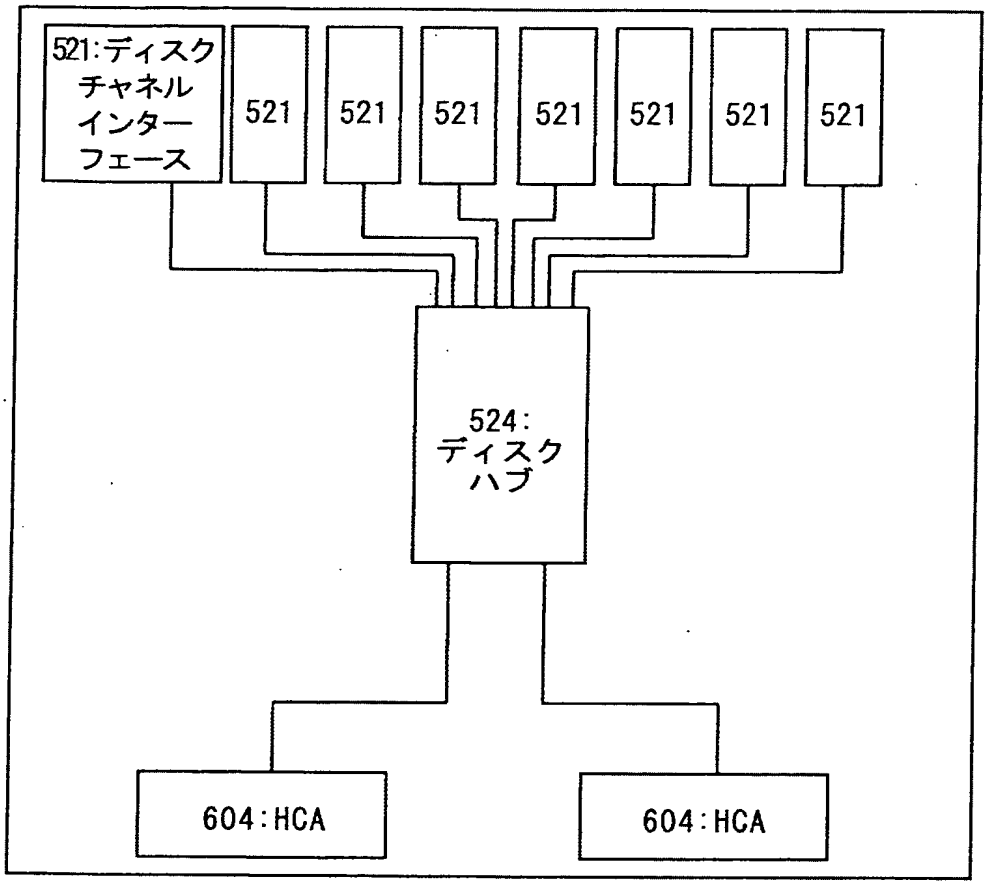
【図１７】

図　１７


511:ホスト
チャネル
インター
フェース
511
511
511
511
511
511
511
514:
ホストハブ
603:HCA
603:HCA

【図１８】

図　１８


521:ディスク
チャネル
インター
フェース
521
521
521
521
521
521
521
524:
ディスク
ハブ
604:HCA
604:HCA

【書類名】要約書
【要約】
【課題】　高い転送効率と高いアプリケーション処理効率を同時に実現するリライアブルデータ転送方法、および、これを用いたディスク制御装置を提供する。
【解決手段】　イニシエータからターゲットにデータ転送を行う際、前記ターゲットで受信されたデータの妥当性を前記データに付加されているエラーチェックコードを用いて確認した上、前記妥当性を前記ターゲットから前記イニシエータに転送ステータスとして返送し、前記転送ステータスにより前記データ転送の際に転送エラーの発生していることが判明した場合、前記イニシエータから前記ターゲットに前記データの再送を行うリライアブル転送において、
前記イニシエータから前記ターゲットへの論理レコードのデータ転送方法であって、
前記イニシエータが発行した転送リクエストにより前記論理レコードの転送が正常に前記ターゲットに到着した時点で、前記論理レコードの前記転送リクエストに対応した完了ステータスを前記ターゲット内に存在する完了キューに通知し、
前記論理レコードを複数まとめて一括転送を行い、
前記イニシエータでは、前記一括転送の単位で前記転送ステータスの確認を行い、
前記ターゲットでは、前記論理レコードの前記転送リクエストに対応した完了ステータスを予め定められた一括転送条件に合致した前記論理レコードについて、その正常受信が完了した時点で前記ターゲット内に存在する完了キューに通知することを特徴とする。
【選択図】　図１

特願２００３－３５３２１９

出　願　人　履　歴　情　報

識別番号　　［０００００５１０８］

１．変更年月日　　１９９０年　８月３１日
［変更理由］　　新規登録
住　所　　東京都千代田区神田駿河台４丁目６番地
氏　名　　株式会社日立製作所